Reconstruction of gene regulatory networks from postgenomic data

Adriano Velasque Werhli

Doctor of Philosophy

Institute for Adaptive and Neural Computation

School of Informatics

University of Edinburgh

2007

Abstract

An important problem in systems biology is the inference of biochemical pathways and regulatory networks from postgenomic data. The recent substantial increase in the availability of such data has stimulated the interest in inferring the networks and pathways from the data themselves. The main interests of this thesis are the application, evaluation and the improvement of machine learning methods applied to the reverse engineering of biochemical pathways and networks. The thesis starts with the application of an established method to newly available gene expression data related to the interferon pathway of the human immune system in order to identify active subpathways under different experimental conditions. The thesis continues with the comparative evaluation of various machine learning methods (Relevance networks, Graphical Gaussian Models, Bayesian networks) using observational and interventional data from cytometry experiments as well as simulated data from a gold-standard network. The thesis also extends and improves existing methods to include biological prior knowledge under the Bayesian approach in order to increase the accuracy of the predicted networks and it quantifies to what extent the reconstruction accuracy can be improved in this way.

Acknowledgements

I would like first to acknowledge the strong support of my supervisor Dr. Dirk Husmeier. I have to say that the support goes beyond the time I have spent here in Edinburgh, in fact it started when I had finished my M.Sc. in Brazil and was looking for a PhD. I am very fortunate to have had a truly devoted researcher as my supervisor and I would like to thank him for all our always stimulating meetings and discussions.

I would also like to acknowledge helpful support from my co-supervisors Douglas Armstrong and Jane Hillston, and stimulating discussions with Peter Ghazal.

Apart from my supervisors I would like to thank all the people from BioSS which were all very supportive and made me feel comfortable in all my years here. Special thanks goes to the people who I shared lunch time with through these PhD years: Alex Cook, Ayona, Isthri, Mizan, Stijn, Stephen, Adam, Kuang Lin and Alex Mantzaris. I consider our talks about general culture and differences among our countries and continents to have been enlightening and one of the most valuable cultural experiences in my life. Alex Cook, was more than sharing lunch time while he was a PhD student; he was really very helpful during my start here in Edinburgh. He represented very well the generosity and kindness of the people from here, which is known as "the best wee country in world". Outside BioSS I am very grateful to Marco Grzegorczyk for our joint project and his always helpful and inspirational answers to my questions. During the time of our collaboration Marco was in Dortmund doing his PhD which he has now successfully completed. I hope in the future we can have new fruitful collaborations.

Although I am very far from Brazil I am in debt with gratitude to many people from there. The support and the kindness offered by friends and family was always very important, much more important than they can realize.

I would like to thank my sponsor CAPES for the financial support during the years I was here pursuing my PhD. More than that I would like to give thanks to

the people of Brazil in general. Effectively the people of my country are my real sponsors. It is through their daily hard work that the government of my country is able to send people like me to study abroad.

At last but not least I would like to thank the person who is my biggest supporter, my wife Zuleica. The beginning in a new country was not easy for us and maybe more challenging for her but not for a moment was she hesitant. She was always focused on our aim and supported me throughout our years here. I am in debt with her and I hope I will be able to retribute all the dedication she devoted to me.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Adriano Velasque Werhli*)

Dedicated to the memory of my younger brother Daniel who for reasons that are beyond our understanding departed very young from this life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past few years we have witnessed the fast development of different techniques for the measurement of high throughput biological data. This has shifted the attention of the research community from a reductionist view towards a more complex understanding of molecular biology systems. Following this fast development and the amount of postgenomic data produced, what still needs to be developed are computational and mathematical tools that in turn will provide us with a better understanding of the biological systems which have generated these data.

In this thesis we are specifically interested in the accurate reconstruction of regulatory networks from postgenomic data. In all living organisms biological components work in an orchestrated way to promote development and sustainability and, therefore, they play a pivotal role in all the processes that occur in these organisms. The manner in which these components work harmonically together is through sets of intricate regulatory networks and pathways.

The discovery of biological pathways or regulatory networks opens a wide range of possible applications. For instance the knowledge of disease related pathways can unveil how the disease acts and present novel tentative drug targets. Also, the creation of accurate biological models from discovered regulatory

networks or pathways can help us to predict the responses to disease/infection and can be very useful in the development of new drugs and treatments. Moreover, the discovery of such pathways in plants would also be very beneficial. With new options to combat plants' diseases one can imagine for example a smaller need for the use of pesticides.

The inference of pathways from the data is still in its infancy though. New types of measurements and the abundance of data produced have brought the hope that one would be able to discover entire pathways from these data. Unfortunately, there are various challenges to be tackled. These data and the biological systems that generate the data are often noisy and, furthermore, the biological processes are frequently not completely understood.

In the next section we present a brief introduction to genetic regulatory networks and to two different types of measurement (data) that can be used to their reconstruction. What follows is a section detailing the general organization of the thesis.

1.1 Biological aspects of genetic regulatory networks

The DNA (Deoxyribonucleic acid) is a long polymer of nucleotides[1] that contains the genetic instructions for the development and the proper functioning of all living organisms. The final product of these genetic instructions are proteins.

Proteins play a key role in all living organism and therefore, can be seen as the main functional components within living cells. For instance, many biochemical reactions which are vital to metabolism are catalyzed by enzymes that are proteins. Moreover, proteins are important in cell signalling, immune response

[1]The nucleotides that form the DNA are: adenine (A), cytosine (C), guanine (G) and thymine (T).

and performing structural functions. Proteins are polymers formed by a linear chain of monomers called amino acids. There are 20 naturally occurring amino acids and the sequence of amino acids observed in a protein is defined by a gene. Genes are segments of the DNA that code for proteins. All cells in an organism carry the same DNA, but synthesized proteins can be totally different. This is due to genetic regulation.

The amount of synthesized protein is regulated by control mechanisms at different stages: transcription, RNA splicing, translation and post-translational modifications. These aforementioned processes together form the core of the so called *Central dogma of molecular biology* (Watson and Crick, 1958; Crick, 1970).

The process of synthesizing proteins from DNA inside a cell can be summarized in a very simplistic way as:

1. Begin with a DNA strand.

2. Transcription: The process of building an RNA copy of a coding DNA sequence. This process starts when one or more transcription factors (TF) bind to a *cis*-regulatory domain of the gene.

3. Translation: The process of matching amino acids to corresponding sets of three bases (codons). During translation messenger RNA (mRNA) sequences are used to manufacture proteins. Translation occurs at special structures in the cell called ribosomes. Ribosomes are the "factories" where RNA is used to manufacture proteins.

4. Post-translational modifications: These are modifications that occur in proteins after they are released from the ribosomes.

5. Finishes with a new protein.

In summary, the information stored in the DNA is processed in the cell machinery and the resulting product are specific types of proteins.

A Genetic Regulatory Network (GRN) is a set of genes (segments of the DNA which code for proteins) that interact with each other in an organism. In a GRN genes control the expression of other genes. In other words, genes control by how much other genes in the network are transcribed into mRNA. Note that not all the genes in network interact with all the other genes. Usually a gene controls only a subset of genes in the network and is itself controlled only by a subset of other genes. Self feedback mechanisms are known to exist in GRNs and hence a gene can regulate itself. The control that a gene performs in other genes (or itself) is indirectly achieved through the RNA and protein expressions. Even though genes do not interact directly with each other, their products (synthesized proteins) in conjunction with other components of the cell regulate the expression of genes in the network. Therefore, this very complex network, which involves many components and steps, is simplified to a model network where the intermediate components are not taken into account, the so called GRN.

The main goal is to understand the relationships among all these components within a cell and how they respond to different challenges. This requires all cell components to be measured at the same time but unfortunately, despite all the recent innovation in molecular biology measurements, this is not yet achievable.

The most common type of postgenomic data available is gene expression data (mRNA concentrations) from microarray experiments and, to a lessen extent, protein activities from flow cytometry experiments. Although these measurements do not cover the whole set of components within a cell they are still useful for the discovery of regulatory networks. Using only this type of data the genetic regulatory network can be seen as a network where the nodes are the genes, the outputs of the nodes are the mRNA concentrations or the protein activities and the inputs to the nodes are the TFs (proteins that start transcription of a gene). The edges in this network represent the set of dependencies and independencies

among the genes (nodes) in the network. The direction of an edge indicates putative causality among two genes. For instance if a gene A is linked to a gene B (A⟶B) it means that the protein produced by A has an influence on the amount of protein that is produced by gene B. Thus we can say that gene A regulates gene B. As mentioned before the interaction between two genes is in fact mediated by many other regulatory events and, hence, we say 'putative causal relationship' instead of 'causal relationship'. See Section 3.2.4 for a discussion about causal networks.

The development of new high throughput molecular biological experiments has produced large quantities of data and, thus, increased the attempts of the reconstruction of GRNs from these data. Among these new experiments microarray technology is one of the most important. With microarray experiments it is possible to measure in parallel the expression profiles of thousands of genes in an organism. This possibility has brought the hope that it would be feasible to reverse engineer GRNs from the data. To-date various methods have been applied to these data in order to reconstruct the GRNs, yet the results have been very modest so far.

Brief introductions to microarray and to flow cytometry technologies are presented in the next section.

1.2 Measuring gene expression and protein activities

1.2.1 Microarrays

In the last few years there has been a great increase in the availability of molecular biological data. The measurement of gene expression using microarrays is one of the more successful techniques among the many methods developed. The seminal

paper on microarray technology is Schena et al. (1995).

Microarrays are nowadays a widely used method that permits the expression profiling of thousands of genes at the same time. In general small amounts of thousands of gene sequences are placed by robotic machines in pre-determined spots of a microscope slide that are called probes. As we discussed in the previous section when a certain gene is active inside a living cell it produces mRNA which in turn is used to produce proteins. If this produced mRNA is complementary to one of the gene sequences placed on the probes it will bind to the corresponding spot. In order to measure the expression of genes in a given cell, the mRNA has first to be collected from the cell and labelled with a fluorescent dye. The labelled mRNA is then placed onto the slide where it will attach to its complementary gene sequence. With a special scanner it is possible to measure the fluorescence of the spots on the slide. Active genes will produce more mRNA, which will attach to the DNA on the microarray producing brighter areas. Spots that are not bright indicate that their genes are not active. The brightness of the spots, measured with the laser scanner, produces measurements which are proportional to the concentration of mRNA. There are two main types of microarrays. One is the spotted microarray where two different experimental conditions (each with its own label) are hybridized to one array. With this fabrication method only relative gene expression values can be estimated. The other type of microarray is the oligonucleotide array where each different condition is hybridized to one array. With this fabrication method it is possible to estimate the absolute values of gene expression. The raw data produced by microarray experiments should go through statistical analysis in order to distinguish genuine biological variation from experimental variation artifacts. The statistical analysis of microarray data is a very active field of research which deals, among others, with the normalization and the significance of microarray measurements. The normalization of microarrays experiments aims to remove sources of variations other than caused by the

biological system itself, thus, the different experimental conditions measured can be fairly compared.

When inferring GRNs from microarray data we need to make one very strong assumption. Effectively microarray experiments provide a measure of the mRNA concentration and this is assumed to be proportional to the protein activity. However, when inferring GRNs we are ultimately interested in protein activities since these are the variables which are likely to influence the other variables in our system. This is because proteins (TFs) are the elements which interact to regulate genes that in turn produce mRNA which is translated to form other proteins. The assumption that the mRNA concentrations are proportional to the protein activities may not hold true in various biological systems due to post-translational modifications that occur after a protein is produced. In order to solve this problem several authors try to infer the activity level of known regulator proteins (TFs) from microarray experiments combined with other sources of molecular data, see for instance Pournara and Wernisch (2007); Sabatti and James (2005).

1.2.2 Flow cytometry

Flow cytometry (Herzenberg et al., 2002; Perez and Nolan, 2002) can measure different parameters in particles and cells using the principles of light scattering, light excitation, and emission of fluorochrome molecules. Particles are hydrodynamically focused on a laser beam and only one particle at a time is presented to the laser beam. Fluorescent chemicals present in the particle (naturally or attached as labels) are excited by the laser, emitting light themselves. This light is measured by detectors and from it, it is possible to gather various types of information about the particle.

Flow cytometers can measure a variety of parameters and of particular interest for the reverse engineering of GRNs is their ability to measure protein expression. While microarrays enable the measurement of thousands of gene ex-

pression profiles simultaneously with usually very few samples, flow cytometry enables the measurement of very few genes within thousands of samples. Flow cytometers can measure up to 18 different parameters (colours) at the same time but the number of measured parameters is not limited and can continue increasing (Bonetta, 2005). In Sachs et al. (2005) flow cytometry was used to measure protein concentrations of 11 proteins. One of the main advantages of flow cytometry is that the variable measured is protein concentration and therefore, we do not need to make the assumption that the mRNA levels are proportional to the protein activities as we make when using microarrays. We still need to assume that the protein concentrations are proportional to the protein activities though.

1.2.3 Organization of the thesis

This thesis can be roughly divided into two major parts. Firstly different methods for the reconstruction of regulatory networks are compared. Given the diversity of proposed reverse engineering methods, it is important for the systems biology community to obtain a better understanding of their relative strengths and weaknesses. The comparison uses both simulated and real data. The use of simulated data is very important as it makes it possible to evaluate the methods' performance given that for this case the true result is known. Furthermore, the use of active interventions is also investigated and the impact of its use in the algorithms' performance is quantified.

The second part is related to the integration of different sources of data or, as we call it, different sources of information with the expression data. Much effort is being put nowadays into investigating methods that are able to use different sources of information. The reason is that there is a huge amount of accumulated knowledge about biological systems but this knowledge is often the result of various different experiments. If it is possible to use all this knowledge together one would expect that the discovery of regulatory networks would be more re-

liable and faster. In this thesis we improve and extend an existing method for integrating other sources of knowledge with expression data. The method enables the integration of more than one source of knowledge at once and each of these sources is associated with a trade-off parameter. The trade-off parameter is learned from the data and indicates how much of the extra knowledge should be used together with gene expression data in order to maximize the regulatory network reconstruction. Using this approach applied to both real and simulated data we show that the reconstruction of the networks is improved, that the method can automatically discard sources of information that are not useful, and that the trade-off parameter learned from the data is close to optimal. Moreover, we explore a version of the same method that, instead of using previous knowledge about the network structure, introduces the idea that networks reconstructed from data of the same biological system obtained under different experimental conditions are likely to share topological features. The method is then applied to simulated and real data and shows consistent improvement over the alternative methods explored.

The thesis is organized as follows:

Chapter 2 reinforces how useful the knowledge of regulatory networks is by presenting a study where a method for the discovery of active subnetworks is applied to a manually curated network. Chapter 3 introduces the statistical and computational methods that are used in this thesis for the learning of genetic regulatory networks. In Chapter 4 a brief introduction to genetic regulatory networks and how we simulate data from a given genetic regulatory are presented. In addition, Chapter 4 also presents the evaluation methods used to quantify the network's reconstruction performance. In Chapter 5 the comparison among different methods for reconstructing genetic regulatory networks is presented. Chapter 6 presents a study where different sources of biological prior knowledge are used in conjunction with expression data for the inference of genetic regulatory

networks. Chapter 7 introduces a method for the integration of data sets obtained from the same biological system challenged with different experimental conditions. And finishing, Chapter 8 presents general conclusions about the work presented in this thesis and discusses some directions for future research in the area of reverse engineering regulatory networks.

Chapter 2

Discovering differentially expressed subnetworks

This chapter presents a simple practical application which shows the use of one known biological pathway and therefore, reinforces how important the automatic discovery of such pathways from data is. Here the network structure is extracted from the literature alone and is used in conjunction with gene expression data from the analysis of macrophage responses to infection. Using gene expression data measured under three different experimental conditions we apply the method of Ideker et al. (2002) to discover subnetworks that are differentially expressed (active) in each of the experimental conditions. The results reveal discrete states of sub-system activity of the Interferon (IFN) pathway and represent a systematic methodology for exploiting biological pathways.

2.1 The manually curated network

Interferons (IFNs), first discovered in 1957, constitute a family of cytokines that play a pivotal role in both the innate and adaptive immune response. While first discovered on the basis of their antiviral properties, they have subsequently been recognized as significant regulators of numerous cellular processes including pro-

liferation, differentiation, apoptosis and antigen presentation. The IFNs may be classified into two types, each signalling through distinct receptors but employing some common signal transducers (Jak, Stat). There are several members of the type I or IFN-α/β superfamily but only one member of the type II family, IFNγ. IFNγ is known as the immune IFN as it is induced by T-cells, neutrophils and natural killer cells and is principally involved in regulation of the immune system and the control of infectious disease. Like most physiological processes, the interferon response is regulated by a pathway of signals transmitted from the receptor to the nucleus. Elucidation of this pathway has engaged a significant proportion of research effort over the past forty years.

Researchers from the Scottish Centre for Genomics Technology and Informatics (GTI) have undertaken a systematic review of the literature relating to components of the IFN pathway using a research synthesis approach. This methodology enabled the definition of interactions and cause-effect relationships in four interconnected functional areas: apoptosis, the interferon regulatory factor (IRF) network, Jak/Stat signalling and antigen presentation. During this process, an attempt was made to curate gene or gene-product interactions which were supported by evidence from at least 3 independent reports and/or laboratories. A particular emphasis was given by the curators on the dependencies of the interactions. It is worth to mention that the process of manually curating the interactions among genes is very lengthy. The ideal scenario would be the automatic discovery of such interactions from data. This would enable the discovery of pathways to be much faster permitting their immediate use by researchers. Unfortunately the discovery of pathways from data is still in its infancy and, hence, this ideal scenario remains far from the reality nowadays.

The data available for the curated network is gene expression measured with microarrays. Therefore, the interest lies specifically in the interactions among genes. In order to have only the variables of interest we extracted, from the whole

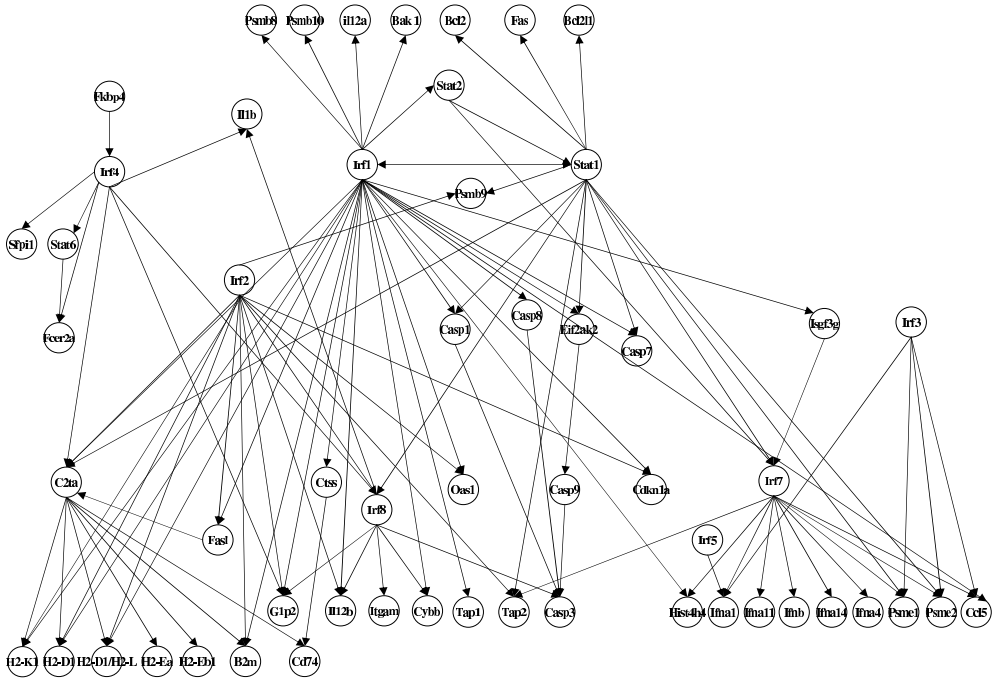


Figure 2.1: **Extracted IFN network.** From the consensus IFN pathway built from the literature we extracted the genetic regulatory network which is presented in this figure. The network is composed by 56 genes and 100 edges.

consensus pathway built from the literature, the genes and their interactions. The extracted genetic regulatory network is constituted by 56 genes and a total of 100 edges connecting these genes. The extracted network is presented in Figure 2.1.

2.2 Gene expression data

To gain insight into the active state of the pathway, microarray analyses were performed on mouse primary Bone Marrow-Derived Macrophages (BMDM) activated with IFNγ and/or challenged with Murine CytoMegaloVirus (MCMV). The three different experimental conditions for which gene expression profiles were measured with microarrays can be summarized as:

- **Infected**: BMDM infected with MCMV.
- **Infected and treated**: BMDM pre-treated with IFNγ and subsequently infected with the virus.

- **Treated**: BMDM activated with physiological levels of IFNγ.

The raw microarray data processing and statistical analysis were performed by GTI. From here onwards when we refer to data we mean the processed data.

2.3 Methodology

In order to verify which subnetworks of the IFNγ genetic network are active in response to different experimental conditions we applied the `jActiveModules` plug-in (Ideker et al., 2002) which is implemented in Cytoscape (Shannon et al., 2003). This method identifies subnetworks that are active, i.e. connected regions of the network that show significant changes in expression under different experimental conditions. The method combines a statistical measure for scoring subnetworks and a search algorithm to find the high scoring subnetworks. The scoring method is based on the p-values which represent the significance of the expression change for each gene. Each p_i, p-value for gene i, is converted to a z-score $z_i = \Phi^{-1}(1 - p_i)$ where Φ^{-1} is the inverse normal cumulative distribution function. To produce an aggregate z-score $Z_{\mathcal{N}}$ for a subnetwork $\mathcal{N}$ with k genes, all the z_i in this subnetwork are summed using the following equation:

$$Z_{\mathcal{N}} = \frac{1}{\sqrt{k}} \sum_{i \in \mathcal{N}}^{k} z_i \tag{2.1}$$

Subnetworks of all sizes are comparable under this scoring system. If z_i are independently drawn from a standard normal distribution, $Z_{\mathcal{N}}$ will also be distributed according to a standard normal independent of k. Note that the variance of sum is the sum of variances for independent random variables.

After calculating the aggregate score it is corrected against random sets of genes, with the same size of the subnetwork. This is to gauge the score against a random allocation of differentially expressed genes and to assess the improvement over what could have been obtained by chance alone. Gene sets of size k are

randomly sampled and their scores $Z_{\mathcal{N}}$ computed. These samples are from the same expression data but independently of the network structure. The computed score values $Z_{\mathcal{N}}$ are then used to produce estimates for the score mean, μ_k, and for the score standard deviation, σ_k. The corrected subnetwork score is give by:

$$s_{\mathcal{N}} = \frac{(Z_{\mathcal{N}} - \mu_k)}{\sigma_k} \tag{2.2}$$

where μ_k and σ_k are respectively the mean and the standard deviation of each cluster of size k.

Having a method to score the subnetworks, a simulated annealing procedure (Kirkpatrick et al., 1983) is applied to find higher scoring subnetworks.

2.3.1 A synthetic example

The purpose of this section is to illustrate the method of Ideker et al. (2002) on a small toy problem which is analytically tractable. For this toy problem it is possible to calculate the scores of such networks analytically and hence, we can examine how the scoring scheme of subnetworks behaves. The small synthetic network, inspired by the real network, is shown in Figure 2.2. It contains 29 white nodes and 3 grey nodes. White nodes have fixed p-value=0.5 and correspond to genes that are not differentially expressed. Grey nodes have fixed p-values=0.01 and correspond to genes that are significantly differentially expressed. In this manner the z-scores of grey nodes, z_G, and white nodes, z_W, are:

$$z_G = \Phi^{-1}(1 - 0.01) = \Phi^{-1}(0.99) = 2.323 \tag{2.3}$$

$$z_W = \Phi^{-1}(1 - 0.5) = \Phi^{-1}(0.5) = 0 \tag{2.4}$$

It is defined that there are k nodes in a subnetwork $\mathcal{N}$, where k is the sum of the number of grey nodes (n_G) and white nodes (n_W): $k = n_G + n_W$. Thus the

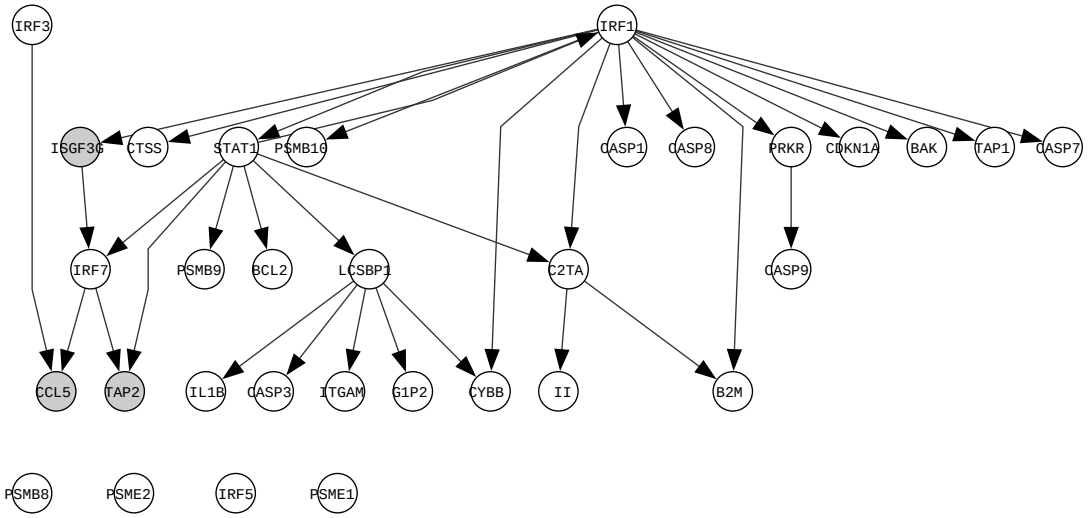


Figure 2.2: **Synthetic example network.** This is the synthetic network which resembles the one extracted from the whole GTI curated pathway. According to the definitions white nodes have p-values of 0.5 and grey nodes have p-values of 0.01.

aggregate score is:

$$Z_{\mathcal{N}} = \frac{n_G z_G + n_W z_W}{\sqrt{n_G + n_W}} \tag{2.5}$$

as $z_W = 0$ according to Equation (2.4), Equation (2.5) reduces to:

$$Z_{\mathcal{N}} = \frac{n_G z_G}{\sqrt{n_G + n_W}} \tag{2.6}$$

It is easy to see that adding extra white nodes, n_W, will only decrease the aggregate score $Z_{\mathcal{N}}$.

Given the structure as in Figure 2.2 we can calculate the scores for different subnetworks $\mathcal{N}_i$. The following subnetworks are defined: $\mathcal{N}_1$={`ISGF3G`}, $\mathcal{N}_2$={`ISGF3G,IRF7`}, $\mathcal{N}_3$={`ISGF3G,IRF7,CCL5`}, $\mathcal{N}_4$={`ISGF3G,IRF7,CCL5,TAP2`} and we continue to add further nodes but note that they are all white nodes. Since the z-scores of all white nodes are $z_W = 0$, according to Equation (2.4), the actual identities of the added white nodes are not relevant.

Considering the subnetwork $\mathcal{N}_1$, we will have its score $Z_{\mathcal{N}_1}$:

$$Z_{\mathcal{N}_1} = \frac{z_G}{\sqrt{1}} = z_G \tag{2.7}$$

and for the subnetwork $\mathcal{N}_2$ the score is:

$$Z_{\mathcal{N}_2} = \frac{z_G + z_W}{\sqrt{2}} = \frac{zG}{\sqrt{2}} \tag{2.8}$$

and for $\mathcal{N}_3$:

$$Z_{\mathcal{N}_3} = \frac{2z_G + z_W}{\sqrt{3}} = \frac{2zG}{\sqrt{3}} \tag{2.9}$$

and for $\mathcal{N}_4$:

$$Z_{\mathcal{N}_4} = \frac{3z_G + z_W}{\sqrt{4}} = \frac{3z_G + z_W}{2} = \frac{3zG}{2} \tag{2.10}$$

and so on for the subsequent networks with included white nodes, which will only decrease the aggregate scores.

The aggregate scores should be calibrated against the background distribution as defined in Equation (2.2). In the synthetic example the mean for a subnetwork $\mathcal{N}_i$ with size k is:

$$\mu_k = \sum_{n_G=0}^{k} P(n_G, k) Z(n_G, k) \tag{2.11}$$

and the variance is:

$$\sigma_k^2 = \sum_{n_G=0}^{k} P(n_G, k)(Z(n_G, k) - \mu_k)^2 \tag{2.12}$$

where the first term $P(n_G, k)$ is the probability of having n_G grey nodes in a subnetwork of size k, which is a binomial distribution:

$$\begin{aligned} P(n_G, k) &= \binom{k}{n_G} q^{n_G} (1-q)^{k-n_G} \\ &= \frac{k!}{n_G!(k-n_G)!} q^{n_G} (1-q)^{k-n_G} \end{aligned} \tag{2.13}$$

where q is the probability of getting a grey node from the entire population. In our case we have a total of 32 nodes 3 of which are grey, hence $q = \frac{3}{32}$. The second term $Z(n_G, k)$ is the aggregate score for n_G grey nodes in a subnetwork of size k.

$$Z(n_G, k) = \frac{n_G z_G}{\sqrt{k}} \tag{2.14}$$

The mean and variance are then:

$$\mu_k = \sum_{n_G=0}^{k} \frac{k!}{n_G!(k-n_G)!} q^{n_G}(1-q)^{k-n_G} \left(\frac{n_G z_G}{\sqrt{k}} \right) \tag{2.15}$$

$$\sigma_k^2 = \sum_{n_G=0}^{k} \frac{k!}{n_G!(k-n_G)!} q^{n_G}(1-q)^{k-n_G} \left[\left(\frac{n_G z_G}{\sqrt{k}} \right) - \mu_k \right]^2 \tag{2.16}$$

Having these equations we can analytically calculate the scores. Figure 2.3 shows the plot of the scores $Z_{\mathcal{N}_i}$ and the corrected scores $S_{\mathcal{N}_i}$. These scores were calculated for the subnetworks: $\mathcal{N}_1$, $\mathcal{N}_2$, $\mathcal{N}_3$, $\mathcal{N}_4$, and then for the following subnetworks, just adding white nodes. In general an analytical calculation presented here will not be tractable, and the algorithm therefore uses an MCMC scheme as described in Ideker et al. (2002).

In the toy problem the grey nodes were chosen in a way that one of the main features of the method (Ideker et al., 2002) can be highlighted. This main feature is the ability of the method in adding structurally important nodes to the subnetworks even if they are not differentially expressed. In our example we can imagine that the node `IRF7` is a transcription factor subject to post-translational modifications. It is structurally important since it intermediates the interaction of three highly differentially expressed genes (grey nodes) but it is not amenable to microarray experiments and therefore, is not differentially expressed. As can be seen from the results the subnetwork $\mathcal{N}_4$ has the highest score. This subnetwork includes the three highly differentially expressed grey nodes and also the non-differentially expressed `IRF7` node that connects them.

The toy problem exemplifies how the method works and shows a very important feature of the method in action. We have also applied the software of Ideker et al. (2002) to the toy problem data set with various different parameters and initializations and the highest scoring subnetwork found was consistently the correct one (results not shown).

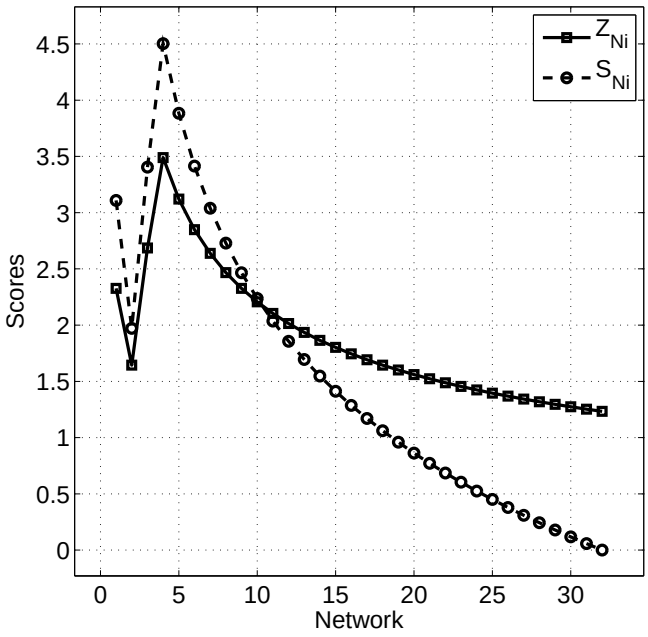


Figure 2.3: **Scores for different subnetworks.** On the horizontal axis are the subnetworks, $\mathcal{N}_1$, $\mathcal{N}_2$, $\mathcal{N}_3$, $\mathcal{N}_4$ and so on. The vertical axis shows the aggregate score $Z_{\mathcal{N}_i}$ and the corrected score $S_{\mathcal{N}_i}$ of these subnetworks.

2.4 Application to real data

We applied the method of Ideker et al. (2002) to the microarray data set provided by GTI. As mentioned before the data was statistically pre-processed at GTI and we used the expression values and significance values (p-values) as provided by them. We set the parameters of the simulated annealing scheme as follows: `Start temperature=100; End temperature=0.001; Number of interactions=`10^6.

For each of the three experimental conditions we ran the simulated annealing from 10 different initializations. Therefore, for a given condition we found 10 subnetworks, one from each different initialization, and these subnetworks were consistent. As for a particular condition the algorithm finds consistently the same subnetwork and we present this network as the highest scoring one. The active subnetworks for each of the conditions are presented graphically in Figure 2.4. The genes that are active in each of the found subnetworks are presented in Table 2.1.

2.5 Results and discussion

In the present study, the consensus diagram was used to experimentally investigate pathway behaviour by measuring changes in node gene expression as a consequence of IFNγ and/or viral infection in macrophages. Gene expression alterations in pathway components were assessed by microarray analyses. Common amongst the three different treatments were four central hubs which were included in the subnetworks (Stat1, Irf1, Irf7 and C2ta). However, it is likely that in the context of the virus vs. IFN stimulated pathways, the activation of these hubs have arisen from different signalling pathways.

While the majority of nodes appearing in the active subnetwork were active in any of the three treatments (infected, infected and treated, treated), some nodes were unique to the presence of virus which were not active in the case of IFN treatment alone. This probably reflects the ability of MCMV to stimulate a type I IFN response resulting in the activation of Interferon Stimulation Response Element (ISRE)-regulated antiviral genes such as Eif2ak2 and effects on cell growth and regulation such as Cdkn1a (Sing et al., 2006). It is notable that the one node unique to the infected and treated condition consisted of the transcription factor Irf8. This protein downregulates the expression of a number of IFN-inducible genes and may represent a feedback mechanism to downregulate the activation state of the macrophage following exposure to both type I (as a result of MCMV infection) and type II IFN signals. On the other hand, Irf8 also plays an important role in macrophage activation, particularly in the context of providing protection against intracellular pathogens such as *Toxoplasma gondii* and *Leishmania donovani* (Sing et al., 2006).

IFNγ activates macrophages and exposure to this cytokine results in the cell adopting a highly efficient antigen presentation phenotype. As a result, macrophages play a key defensive role in protecting against pathogens such as MCMV. We have applied the method of Ideker et al. (2002) for analysing the

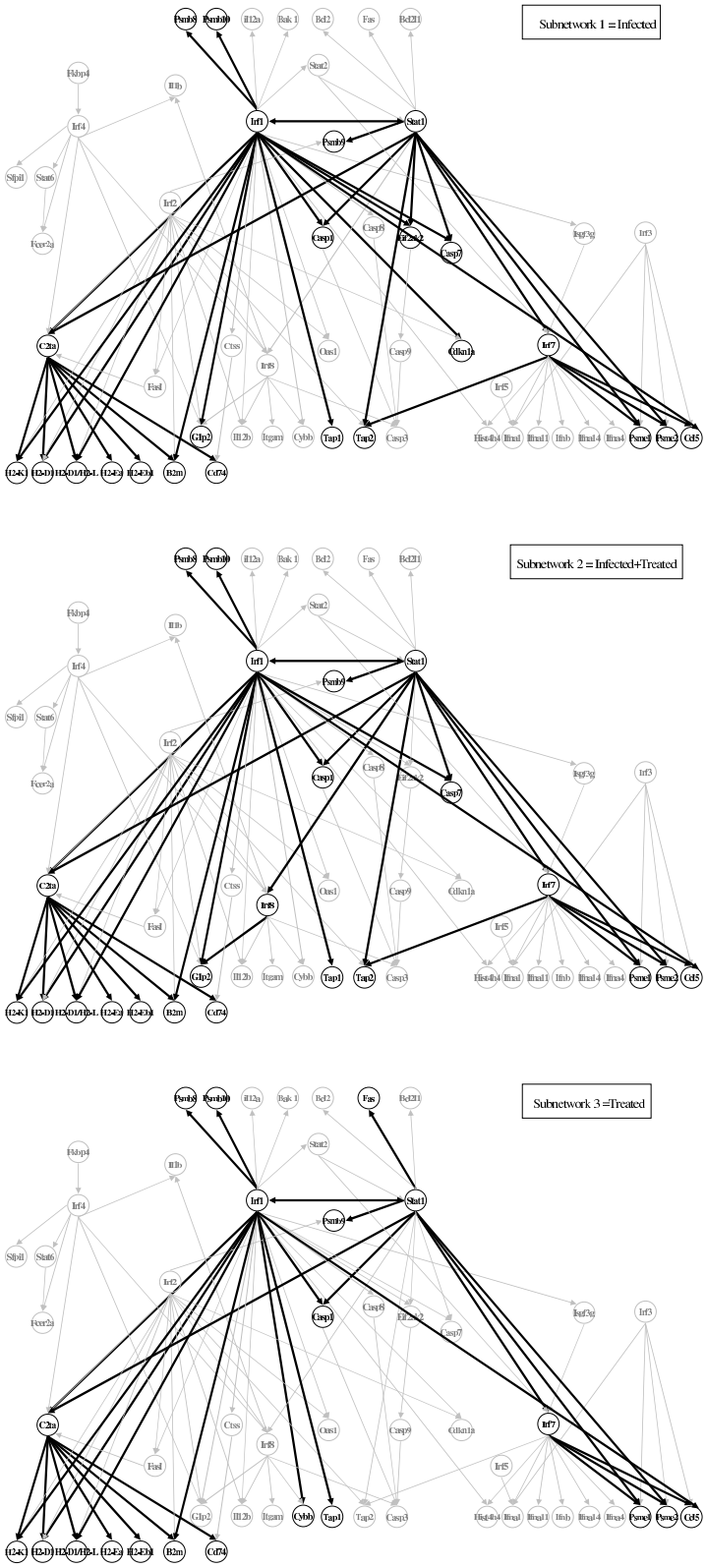


Figure 2.4: **Inferred subnetworks.** Each of these panels presents highlighted the subnetwork found for the different experimental conditions. The top panel shows the subnetwork found when considering the infected condition. The middle panel shows the subnetwork found for the infection and interferonγ treatment and the bottom panel shows the subnetwork found for treatment only.

IFNg genetic network		
Infected	**Infected+treated**	**treated**
B2m	B2m	B2m
C2ta	C2ta	C2ta
Casp1	Casp1	Casp1
Ccl5	Ccl5	Ccl5
Cd74	Cd74	Cd74
H2-D1	H2-D1	H2-D1
H2-D1/H2-L	H2-D1/H2-L	H2-D1/H2-L
H2-Ea	H2-Ea	H2-Ea
H2-Eb1	H2-Eb1	H2-Eb1
H2-K1	H2-K1	H2-K1
Irf1	Irf1	Irf1
Irf7	Irf7	Irf7
Psmb10	Psmb10	Psmb10
Psmb8	Psmb8	Psmb8
Psmb9	Psmb9	Psmb9
Psme1	Psme1	Psme1
Psme2	Psme2	Psme2
Stat1	Stat1	Stat1
Tap1	Tap1	Tap1
Casp7	Casp7	
G1p2	G1p2	
Tap2	Tap2	
Cdkn1a		
Eif2ak2		
	Irf8	
		Cybb
		Fas

Table 2.1: **Resulting subnetworks for INFg genetic network.** In this table are shown the genes that compose the highest scoring subnetwork found by the simulated annealing algorithm to each experimental condition.

activity of the IFN pathway upon infection. Notably, these studies showed that the emergent expression changes due to the various macrophage perturbations mapped to discrete and specific sub-systems of the IFN pathway. Specifically, the activity of sub-systems involving some, but not all, components of apoptosis (Casp7, G1p2) and antigen presentation (Tap2) are altered. Our results provide evidence for discrete sub-system activity of the IFN pathway and support the notion that the pathway can adopt a number of different states. However, because not all the pathways are present in our network some of the interactions that arise due to the activity of these other pathways cannot be distinguished from the interactions that arise from the given pathway.

The present study represents a step towards a comprehensive picture of the IFN pathway and serves as a foundation for understanding the molecular circuitry of a key cell-injury response pathway and its role in health and disease. In the future with a more comprehensive coverage of the IFN pathway and with the presence of other pathways it will be easier to identify and clearly separate which are the different subsystems acting in the presence of different challenges. From such an exercise, it should be possible to generate a more comprehensive view of one of the most intensively studied and fundamental biological pathways of the immune system.

Chapter 3

Statistical Methods for Inferring Gene Regulatory Networks

3.1 Introduction

In the last few years, several methods to the reconstruction of regulatory networks and biochemical pathways from data have been proposed. These methods were reviewed for example in De Jong (2002); D'haeseleer et al. (2000).

Differential equations are the most refined mathematical method to describe biophysical processes. They can describe, for example, the intra-cellular processes of transcription factor binding, diffusion, and RNA degradation; see, for instance, Chen et al. (1999). Such detailed descriptions of the dynamics are essential to an accurate understanding of regulatory networks but they require substantial prior knowledge about the system. For instance it is necessary to specify how the entities of the system relate with each other and all the parameters of the biochemical reactions. Although differential equations are the most accurate way of representing regulatory networks their use is limited by the necessity of substantial prior knowledge about the system they are representing. At the other extreme is the coarse grain approach of clustering which has been widely applied to the

analysis of microarray gene expression data D'haeseleer et al. (2000); Eisen et al. (1998). Clustering is computationally very cheap to extract qualitative information about co-expression, but it is not powerful to provide the inference of the detailed structure of the underlying biochemical signalling pathways.

A promising compromise between these two extremes are machine learning methods that allow interactions between the nodes in the network to be represented in an abstract way - without the level of detail of the underlying pathways described by differential equation models - and to infer these interactions from data in a systems context, that is, distinguishing direct interactions from indirect interactions that are mediated by other nodes in the domain. This chapter provides a review of various machine learning methods that have been applied to the reconstruction of gene regulatory networks. We address the issue of practical viability of these approaches in Chapter 5, where details of a comparative evaluation study using benchmark data from a widely-studied model network are presented.

3.2 Bayesian Networks

Bayesian Networks (BNs) are a combination of probability theory and graph theory. They are very useful to represent probabilistic relationships between multiple interacting entities. Their nodes represent random variables and its arcs represent dependencies between these random variables. Formally a BN is fully specified by a graphical structure $\mathcal{M}$, a family of conditional probability distributions $\mathcal{F}$ and their parameters $\mathbf{q}$.

The graphical structure $\mathcal{M}$ is a directed acyclic graph (DAG). DAGs are graphics that have only directed edges between nodes and have no directed cycles. They indicate conditional dependence relations between nodes through their edges. The family of conditional probability distributions $\mathcal{F}$ and their parameters

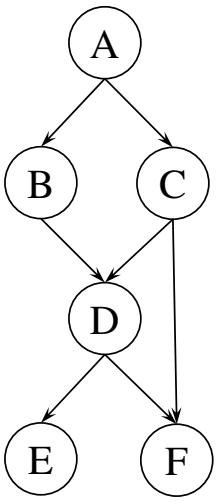


Figure 3.1: Example of Bayesian Network. This figure presents a Bayesian Network example composed of the set of nodes $N=\{A, B, C, D, E, F\}$ and edges $\mathcal{E}=\{(A, B),(A, C),(B, D),(C, D),(D, E),(D, F),(C, F)\}$. Applying the independence relationships depicted by the graph we can write the joint probability $P(A, B, C, D, E, F)$ as $P(A)P(B|A)P(C|A)P(D|B, C)P(E|D)P(F|D, C)$.

$\mathbf{q}$ specify the functional form of the conditional probabilities associated with the edges, that is, they indicate the nature of the interactions between nodes and the intensity of these interactions.

3.2.1 Bayesian Networks Structure

Figure 3.1 shows a hypothetical Bayesian network. This network is constituted by the set of nodes $N=\{A, B, C, D, E, F\}$ where the set of dependencies between them is represented by the set of directed edges $\mathcal{E}=\{(A, B),(A, C),(B, D),(C, D),(D, E),(D, F),(C, F)\}$. If we have a directed edge from a node A to a node B, then A is called *parent* of B, and B called the *child* or *descendant* of A.

A BN is characterized by a simple and unique rule for expanding the joint probability in terms of simpler conditional probabilities. This follows the local Markov property: *A node is conditionally independent of its non descendants*

given its parents. Thus we can write the chain rule or factorization rule:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \pi_{\mathcal{M}}(X_i)) \tag{3.1}$$

Note that we use the same symbols to represent the nodes and the the random variables that they represent, e.g. (X_i). In the same way the set of parent nodes and the random variables that they represent also have the same symbol, e.g. $(\pi_{\mathcal{M}}(X_i))$. Thus in Equation (3.1) $X_1, X_2, \ldots, X_n$ are random variables represented by nodes $X_i \in 1, \ldots, n$ and $\pi_{\mathcal{M}}(X_i)$ is the set of random variables represented by the set of nodes $\pi_{\mathcal{M}}(X_i)$ which are the parents of node X_i in the model $\mathcal{M}$.

If we apply Equation 3.1 to the BN in Figure 3.1, we obtain the factorization

$$P(A, B, C, D, E, F) = P(A)P(B|A)P(C|A)P(D|B, C)P(E|D)P(F|D, C) \tag{3.2}$$

A graph provides a scheme for expanding the joint probability into a product of lower complexity conditional probabilities like in Equation (3.2). In other words, following our example in Figure 3.1, we apply the chain rule, Equation (3.1), and we have the product as specified in Equation (3.2). To specify the complete joint distribution it is still necessary to determine each of the conditional probabilities in the product form, Equation (3.2). As pointed out in Friedman et al. (2000) to represent these families $\mathcal{F}$ of conditional distributions it is possible to choose from different representations for example Gaussian and Multinomial. This choice will depend on the type of variable we are dealing with, continuous variables or discrete variables. Let us define that the set of parents of a variable X_i is $\pi_{\mathcal{M}}(X_i)$ and look at each case:

- Discrete variables: This is the case where each of the X_i and its parents $\pi_{\mathcal{M}}(X_i)$ takes discrete values from a finite set. In this case the conditional probabilities $P(X_i|\pi_{\mathcal{M}}(X_i))$ can be represented as a table that specifies the probabilities of values for X_i for each joint assignment to $\pi_{\mathcal{M}}(X_i)$. This representation can describe any discrete conditional distribution.

- Continuous variables: When the variable X_i and its parents $\pi_{\mathcal{M}}(X_i)$ are real valued there is no representation that can represent all possible densities. The natural choice for this case is to use linear Gaussian conditional densities, so the conditional density of X_i given its parents $\pi_{\mathcal{M}}(X_i)$ is:

$$P(X_i|\pi_{\mathcal{M}}(X_i)) \sim N(\mu_0 + \sum_{k \in \pi_{\mathcal{M}}(X_i)} b_{ik} X_k, \sigma^2) \tag{3.3}$$

  This means that X_i is normally distributed around a mean that depends linearly on the weighted values of its parents, $\sum_{k \in \pi_{\mathcal{M}}(X_i)} b_{ik} X_k$, and on the unconditional mean μ_0. Here the sum index, $k \in \pi_{\mathcal{M}}(X_i)$, means that the sum extends over all the individual k nodes which compose the parent set.

Each of these representations have advantages and drawbacks. In the linear Gaussian representation there is no need to discretize the data, but it can only handle dependencies that are close to linear. The multinomial representation can capture dependencies that are non-linear, but it is necessary to discretize the data causing some loss of information. These two ways of assigning the conditional probabilities will be discussed in more detail in Section 3.2.5.

3.2.2 Learning Bayesian Networks

Learning a Bayesian Network means that we want to devise a BN from a given set of training data $\mathcal{D}$. At the end we want to have a DAG with a set of parametrized conditional probabilities that better explains the data. In order to learn a Bayesian Network it is not necessary to use Bayesian learning, but we will focus on this approach. Learning a BN is a two stage process where first we learn the structure, the edges that connect our entities. Second we learn the parameter sets associated with these edges and whether the relationships between these entities are activating or inhibitory as well as its intensity. Defining that $\mathbb{M}$ is the space of all models, the first goal is to find a model $\mathcal{M}^* \in \mathbb{M}$ that is most

supported by the data $\mathcal{D}$:

$$\mathcal{M}^* = \text{argmax}_{\mathcal{M}} \{\mathcal{P}(\mathcal{M}|\mathcal{D})\} \tag{3.4}$$

Having the best structure $\mathcal{M}^*$ and the data $\mathcal{D}$, we can now find the best parameters $\mathbf{q}$:

$$\mathbf{q} = \text{argmax}_{\mathbf{q}} \{P(\mathbf{q}|\mathcal{M}^*, \mathcal{D})\} \tag{3.5}$$

If we apply Bayes' rule to Equation (3.4) we get:

$$P(\mathcal{M}|\mathcal{D}) \propto P(\mathcal{D}|\mathcal{M})P(\mathcal{M}) \tag{3.6}$$

where the marginal likelihood implies an integration over the whole parameter space:

$$P(\mathcal{D}|\mathcal{M}) = \int P(\mathcal{D}|\mathbf{q}, \mathcal{M})P(\mathbf{q}|\mathcal{M})d\mathbf{q} \tag{3.7}$$

The integral in Equation (3.7) is analytically tractable when the data is complete and the prior $P(\mathbf{q}|\mathcal{M})$ and the likelihood $P(\mathcal{D}|\mathbf{q}, \mathcal{M})$ satisfies certain regularity conditions (Heckerman, 1994, 1995). In Section 3.2.5 following Heckerman (1994, 1995) we present two ways of solving this integral.

Now we turn our attention to the term $P(\mathcal{M})$ from Equation (3.6). This term is the prior over structures. The simplest choice of prior is the one which is uniform over structures. Given the whole space of models $\mathbb{M}$ the uniform prior over structures is defined by:

$$P(\mathcal{M}) = \frac{1}{|\mathbb{M}|} \tag{3.8}$$

where $|\mathbb{M}|$ denotes the number of possible models.

Another option for the prior over structures is to consider it uniform over parent cardinalities. Deciding that a node X_i has $|\pi_{\mathcal{M}}(X_i))|$ parents there are $\binom{n-1}{|\pi_{\mathcal{M}}(X_i))|}$ possible parent sets, where n is the total number of nodes. If we propose uniformly from these, the prior is:

$$P(\mathcal{M}) = \frac{1}{Z} \prod_{i=1}^{n} \binom{n-1}{|\pi_{\mathcal{M}}(X_i))|}^{-1} \tag{3.9}$$

where Z is a normalizing constant.

One of the important properties of these priors is that they satisfy the structure modularity. In this way it is possible to decompose the prior in a product where each term corresponds to one family in $\mathcal{M}$. If we define $\rho(X_i, \pi_{\mathcal{M}}(X_i))$ to be the contribution to the prior from the structure formed by node X_i and its parents $\pi_{\mathcal{M}}(X_i)$, the prior of the whole model can be written as:

$$P(\mathcal{M}) = \prod_{i=1}^{n} \rho(X_i, \pi_{\mathcal{M}}(X_i)) \tag{3.10}$$

Now we examine the term $P(\mathcal{D}|\mathcal{M})$ from Equation (3.6). If the regularity conditions discussed in Heckerman (1994, 1995) are satisfied, the marginal likelihood $P(\mathcal{D}|\mathcal{M})$ can be factorised as:

$$P(\mathcal{D}|\mathcal{M}) = \prod_{i=1}^{n} \psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D}) \tag{3.11}$$

Here $\psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D})$ is the score of the structure formed by node X_i and their parents $\pi_{\mathcal{M}}(X_i)$ given the data $\mathcal{D}$. Furthermore if the prior $P(\mathcal{M})$ satisfies the modularity property (3.10) we can write:

$$\begin{aligned} P(\mathcal{M}|\mathcal{D}) &\propto \\ P(\mathcal{D}|\mathcal{M})P(\mathcal{M}) &= \prod_{i=1}^{n} \rho(X_i, \pi_{\mathcal{M}}(X_i))\psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D}) \end{aligned} \tag{3.12}$$

In order to find the model that is best supported by the data one approach is to compute $P(\mathcal{M}|\mathcal{D})$ for all possible structures $\mathcal{M} \in \mathbb{M}$ and choose the one that maximizes $P(\mathcal{M}|\mathcal{D})$. The first problem with this approach is that the number of structures increases rapidly with the number of nodes as we can see in Table 3.1 making an exhaustive search impossible. The second problem is that the typical systems biology data is sparse. Therefore the posterior $P(\mathcal{M}|\mathcal{D})$ is usually diffuse and will not be adequately represented by a single network at the mode; see Figure 3.2 for an example. To overcome these difficulties we resort to an MCMC sampling scheme.

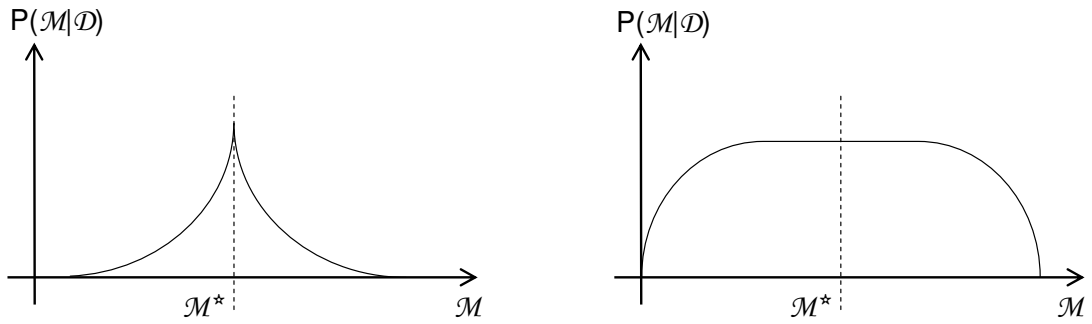


Figure 3.2: **Inference uncertainty.** The vertical axis shows the posterior probability $P(\mathcal{M}|\mathcal{D})$ and the horizontal axis represents the model structure $\mathcal{M}$. The left panel shows the posterior probability for a large and informative data set where the best structure $\mathcal{M}^*$ is very well defined. In the right panel the data set is small and less informative and there are many structures with high scoring posterior probability leading to a large uncertainty about the best structure.

Number of nodes	2	4	6	8	10
Number of topologies	3	543	3.7×10^6	7.8×10^{11}	4.2×10^{18}

Table 3.1: **Number of nodes vs. number of networks.** The number of networks grows super-exponentially with the number of nodes. Taken from Murphy (2001)

3.2.2.1 Sampling Networks with Markov Chain Monte Carlo

When inferring genetic networks from postgenomic data, the data $\mathcal{D}$ is generally sparse and therefore the posterior over the structures $P(\mathcal{M}|\mathcal{D})$ is diffuse, meaning that $P(\mathcal{M}|\mathcal{D})$ will not be properly represented by a single structure $\mathcal{M}^*$. In this case it is more appropriate to sample networks from the posterior probability:

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M})P(\mathcal{M})}{P(\mathcal{D})} = \frac{P(\mathcal{D}|\mathcal{M})P(\mathcal{M})}{\sum_{\mathcal{M}'} P(\mathcal{D}|\mathcal{M}')P(\mathcal{M}')} \tag{3.13}$$

in this way we can obtain a representative sample of high scoring network structures. A direct approach to sampling from $P(\mathcal{M}|\mathcal{D})$ is impossible though, as the denominator in Equation (3.13) is a sum over the whole model space and is intractable. Table 3.1 can give an idea about the size of the sampling space. A solution to this problem is to create a Markov Chain, as was proposed in Metropolis et al. (1953); Hastings (1970) and applied to Bayesian Networks by

Madigan and York (1995). This Markov chain has the following form:

$$P_{n+1}(\mathcal{M}_{\text{new}}) = \sum_{\text{old}} T(\mathcal{M}_{\text{new}})|\mathcal{M}_{\text{old}})P_n(\mathcal{M}_{\text{old}}) \tag{3.14}$$

The transition from one model $\mathcal{M}_{\text{old}}$ into the model $\mathcal{M}_{\text{new}}$, $T(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})$, is represented by the Markov transition matrix T, which is a matrix of transition probabilities. The important feature of a Markov chain is that under the condition of ergodicity[1] the distribution $P_n(\mathcal{M}_{\text{old}})$ converges to a stationary distribution $P_\infty(\mathcal{M}_{\text{old}})$:

$$P_n(\mathcal{M}_{\text{old}}) \stackrel{n\to\infty}{\longrightarrow} P_\infty(\mathcal{M}_{\text{old}}) \tag{3.15}$$

The transition matrix T determines completely this stationary distribution, so we can write:

$$P_\infty(\mathcal{M}_{\text{new}}) = \sum_{\text{old}} T(\mathcal{M}_{\text{new}})|\mathcal{M}_{\text{old}})P_\infty(\mathcal{M}_{\text{old}}) \tag{3.16}$$

and what we need is to design the transition matrix T so that we get the posterior probability as the stationary distribution of the Markov chain, $P(\mathcal{M}|\mathcal{D}) = P_\infty(\mathcal{M})$. If the Markov chain of (3.14) converges to the posterior probability of Equation (3.13) we can write:

$$P_n(\mathcal{M}) \stackrel{n\to\infty}{\longrightarrow} P(\mathcal{M}|\mathcal{D}) \tag{3.17}$$

A sufficient condition for this to be true is the equation of detailed balance:

$$\frac{T(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})}{T(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})} = \frac{P(\mathcal{M}_{\text{new}}|\mathcal{D})}{P(\mathcal{M}_{\text{old}}|\mathcal{D})} = \frac{P(\mathcal{D}|\mathcal{M}_{\text{new}})P(\mathcal{M}_{\text{new}})}{P(\mathcal{D}|\mathcal{M}_{\text{old}})P(\mathcal{M}_{\text{old}})} \tag{3.18}$$

The transition from a structure into another, $\mathcal{M}_{\text{old}} \to \mathcal{M}_{\text{new}}$, consists of two parts, first we propose a new structure with a proposal probability $Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})$ and second we need to accept this new structure with acceptance probability

[1] An ergodic Markov chain is aperiodic and irreducible. An irreducible Markov chain is one in which all states are reachable from all other states. A sufficient test for a aperiodicity is that each state has a "self-loop", meaning that the probability that the next state is the same as the current state is non-zero. In general it is difficult to prove that a Markov chain is ergodic, however ergodicity can be assumed to hold in most real-world applications.

Figure 3.3: **Metropolis-Hastings Algorithm.**

- Start with a initial structure $\mathcal{M}_0$
- Iterate for $i = 1 \dots I$
  1. Obtain a new DAG structure $\mathcal{M}_i$ from the proposal distribution $Q(\mathcal{M}_i|\mathcal{M}_{i-1})$.
  2. Accept the new model with probability $A(\mathcal{M}_i|\mathcal{M}_{i-1})$, given by Equation (3.20), otherwise leave the model unchanged
- Allow the Markov chain to reach stationarity discarding some initial samples. This is the burn-in period. For example discard $\mathcal{M}_1 \dots \mathcal{M}_{I/2}$.
- Compute the expectation values from the MCMC sample $\{\mathcal{M}_{\frac{I}{2}+1} \dots \mathcal{M}_I\}$:
- $\langle f \rangle = \sum_{\mathcal{M}} f(\mathcal{M}) P(\mathcal{M}|\mathcal{D}) \approx \frac{2}{I} \sum_{i=I/2+1}^{I} f(\mathcal{M}_i)$

$A(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})$. The transition probability is then the product of these two probabilities. Inserting this product into Equation (3.18) we obtain the following equation for the acceptance probabilities:

$$\frac{A(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})}{A(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})} = \frac{P(\mathcal{D}|\mathcal{M}_{\text{new}})P(\mathcal{M}_{\text{new}})Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})}{P(\mathcal{D}|\mathcal{M}_{\text{old}})P(\mathcal{M}_{\text{old}})Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})} \tag{3.19}$$

for which a sufficient condition is:

$$A(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}}) = \min\left\{\frac{P(\mathcal{D}|\mathcal{M}_{\text{new}})P(\mathcal{M}_{\text{new}})Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})}{P(\mathcal{D}|\mathcal{M}_{\text{old}})P(\mathcal{M}_{\text{old}})Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})}, 1\right\} \tag{3.20}$$

From these previous derivations we can see that accepting a new configuration $\mathcal{M}_{\text{new}}$ with probability given by Equation (3.20) is the condition to satisfy the Equation of detailed balance (3.18), which guarantees that the Markov chain will converge to the desired posterior distribution of Equation (3.13). The Equation (3.20) is known as the Metropolis-Hastings acceptance criterion (Hastings,

Figure 3.4: **Metropolis Algorithm.**

- Start with a initial structure $\mathcal{M}_0$
- Iterate for $i = 1 \dots I$
    1. Obtain a new structure $\mathcal{M}_i$ from the proposal distribution $Q(\mathcal{M}_i|\mathcal{M}_{i-1})$.
    2. If the new model is not a DAG reject it and go back to the previous step.
    3. Accept the new model with probability $A(\mathcal{M}_i|\mathcal{M}_{i-1})$, given by Equation (3.21), otherwise leave the model unchanged
- Allow the Markov chain to reach stationarity discarding some initial samples. This is the burn-in period. For example discard $\mathcal{M}_1 \dots \mathcal{M}_{I/2}$.
- Compute the expectation values from the MCMC sample $\{\mathcal{M}_{\frac{I}{2}+1} \dots \mathcal{M}_I\}$:
- $\langle f \rangle = \sum_{\mathcal{M}} f(\mathcal{M}) P(\mathcal{M}|\mathcal{D}) \approx \frac{2}{I} \sum_{i=I/2+1}^{I} f(\mathcal{M}_i)$

1970) which is a generalization of the Metropolis (Metropolis et al., 1953) algorithm for proposal distributions Q which are not symmetric. For symmetric proposal distributions Q Equation (3.20) can be rewritten as:

$$A(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}}) = \min\left\{\frac{P(\mathcal{D}|\mathcal{M}_{\text{new}})P(\mathcal{M}_{\text{new}})}{P(\mathcal{D}|\mathcal{M}_{\text{old}})P(\mathcal{M}_{\text{old}})}, 1\right\} \tag{3.21}$$

The MCMC is exact in the limit of an infinitely long Markov chain, if the condition of detailed balance is satisfied and if the Markov chain is ergodic. In practice a bad initialization can slow down the mixing and convergence of the Markov chain. A simple test to check the convergence is to run MCMC with two different initializations and plot the posterior probabilities of the edges against each other. This test, however, is only a necessary condition but not sufficient. The two simulations can reach the same meta-stable equilibrium which is different from the true equilibrium. This test of convergence is discussed in more detail in Section 3.2.2.4.

3.2.2.2 Standard MCMC

The standard MCMC scheme consists in proposing a new structure and accepting the structure according to Equation (3.20). This algorithm is presented in Figure 3.3. The action of proposing a new structure is to propose, at each interaction, one of the basic operations of adding, removing or reversing an edge. These operations are presented in Figure 3.5. As exemplified in Figure 3.5 some of these basic operations can lead to networks that are not allowed due to the presence of directed cyclic structures and these networks need to be discarded.

When computing the acceptance probability according to Equation (3.20) it is necessary to properly calculate the Hastings factor, the ratio between the proposal probabilities, since these are not always symmetric in this case. The asymmetry is a consequence of the different neighbourhood sizes that each of the structures associated with the proposal move can have. We define a neighbour structure as

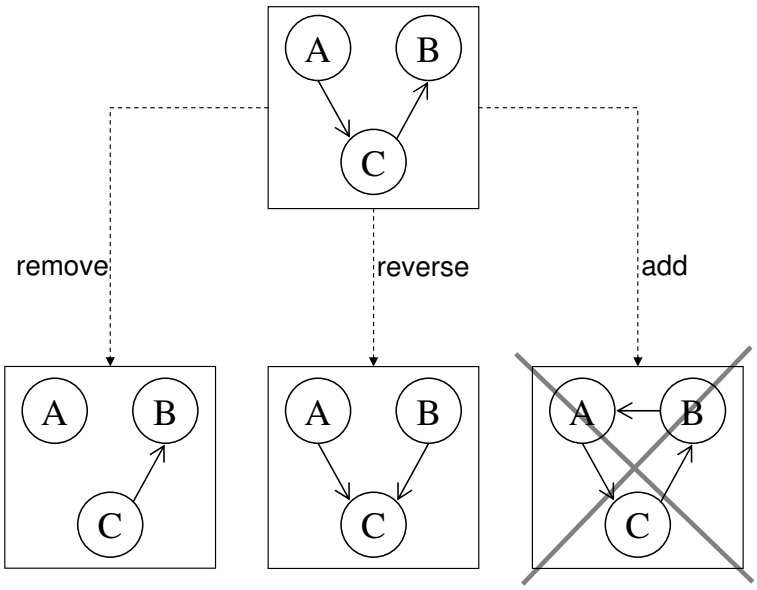


Figure 3.5: **Standard MCMC.** Standard MCMC moves. Here we present the three basic operations that we use to propose a new structure in the standard MCMC. Given the top structure we can propose the removal, the reversion or the addition of an edge. Left subfigure shows the removal of an edge. Note that this operation never leads to invalid structures. The middle subfigure shows the reversal of an edge. This operation can lead to invalid structures. The right subfigure shows the addition of an edge. In this case the proposed structure is invalid since it is not a proper DAG.

any valid DAG structure which can be reached from the current DAG structure with one of the moves presented in Figure 3.5.

Figure 3.6 shows the example of one situation where the proposal probabilities are not symmetric. If we define the number of graphs that are neighbours of the actual structure as $\mathcal{N}(\mathcal{M}_{\text{old}})$ and the number of graphs that are neighbours of the proposed graph as $\mathcal{N}(\mathcal{M}_{\text{new}})$, the Hastings factor will be:

$$\frac{Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})}{Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})} = \frac{\frac{1}{\mathcal{N}(\mathcal{M}_{\text{new}})}}{\frac{1}{\mathcal{N}(\mathcal{M}_{\text{old}})}} = \frac{\mathcal{N}(\mathcal{M}_{\text{old}})}{\mathcal{N}(\mathcal{M}_{\text{new}})} \tag{3.22}$$

It is clear that for properly computing the Hastings factor it is necessary to determine the number of all valid DAGs in the neighbourhoods of the two DAGs involved in the proposal move. In order to avoid the necessity of determining these neighbourhood sizes it is possible to modify the MCMC algorithm causing the proposal probabilities to be symmetric. The modified algorithm proposes

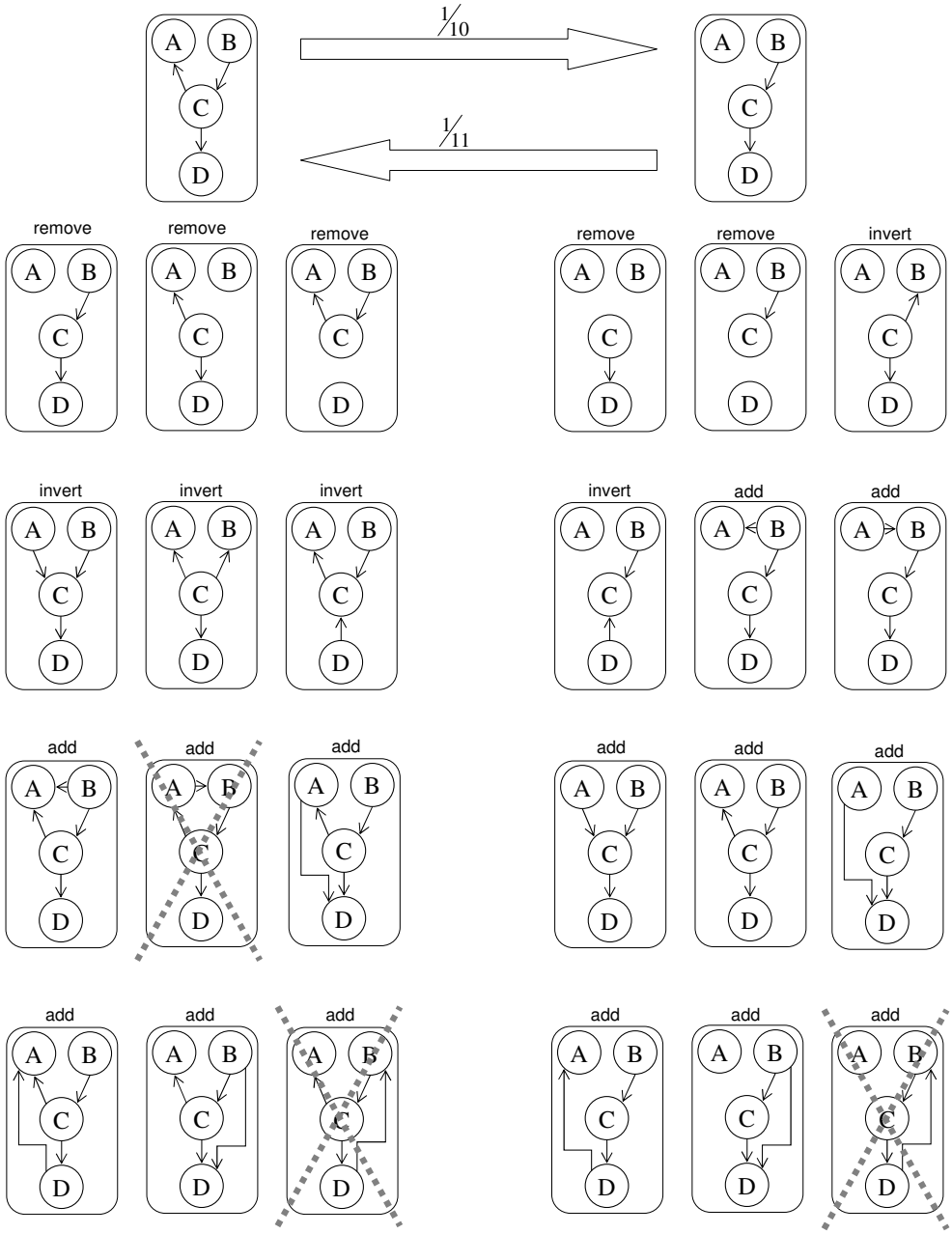


Figure 3.6: **Example of non symmetric proposal probabilities.** The left part of the figure shows a BN and all its possible neighbours along with the operations that give rise to such structures. The right part of the figure shows a proposed BN and its possible neighbours again with the operations that give rise to these structures. Neighbours that are not proper BNs are crossed out. The top large arrow shows the probability of proposing the new structure and the arrow below shows the probability of returning from the proposed structure do the original structure. This example show that the proposal probabilities using the operations of adding, reversing or removing an edge are not always symmetric due to the possibility of different neighbourhood sizes.

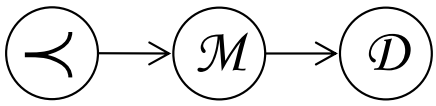


Figure 3.7: **Order MCMC independence relationship.** This figure shows the independence relationship between the imposed ordering in the nodes, $\prec$, the BN, $\mathcal{M}$ and the data $\mathcal{D}$. Therefore the joint probability $P(D, \mathcal{M}, \prec)$ can be written as $P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\prec)P(\prec)$.

graphs that are not proper DAGs and then reject these graphs on the basis of the prior knowledge that they cannot be accepted. By doing this it is possible to use Equation 3.21, the Metropolis criterion, instead of Equation 3.20, the Metropolis-Hastings criterion. The MCMC algorithm for this case is presented in Figure 3.4. Note that in the Metropolis-Hastings algorithm where it is necessary to properly calculate the Hastings ratio, the invalid structures are not proposed. Conversely, in the Metropolis algorithm, the one with symmetric proposal probabilities, the invalid structures are effectively proposed and rejected on the basis of the prior by the algorithm. This possibility to avoid the finding of all neighbours comes at a price though. With this approach many more structures will be rejected, making the acceptance rate decrease and slowing down the convergence of the algorithm.

3.2.2.3 Order MCMC

The order MCMC algorithm was proposed by Friedman and Koller (2003). The principal point of this approach is to focus the attention on a search space of node orders instead of a search space of DAG structures as in the standard MCMC.

A given order, $\prec$, specifies that if $X_i \in \pi_{\mathcal{M}}(X_j)$ then $X_i \prec X_j$. This means that the only nodes that are allowed to be parents of a node X_j are the ones that precede X_j according to the given order $\prec$. Therefore, for a given order all the nodes to the left of a node X_j can be parents of X_j, conversely the nodes to the right are not permitted to be parents of X_j. With this approach, instead of sampling networks structures given the data, the attention is turned to the problem of sampling orders given the data. Given the independence relationship

presented in Figure 3.7 we can calculate $P(\mathcal{D}|\prec)$ as:

$$P(\mathcal{D}|\prec) = \frac{P(D,\prec)}{P(\prec)} \tag{3.23}$$

$$= \frac{\sum_{\mathcal{M}\in\mathbb{M}} P(\mathcal{D},\prec,\mathcal{M})}{P(\prec)} \tag{3.24}$$

$$= \frac{\sum_{\mathcal{M}\in\mathbb{M}} P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\prec)P(\prec)}{P(\prec)} \tag{3.25}$$

$$= \sum_{\mathcal{M}\in\mathbb{M}} P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\prec) \tag{3.26}$$

The final sum over all the possible $\mathcal{M}$ graphs is still exponentially large. The key point is that by imposing an order on the nodes the choice of the families for one node does not add constraints to the choice of families for another node. In other words, because the nodes can only have families that are consistent with a given order the possibility of directed cyclic networks is eliminated. Each node X_i has k possible parent sets or families which are consistent with a given order $\prec$ and we represent such family as $\pi_{\mathcal{M}}(X_i)_{k,\prec}$. Each parent node X_j which is included in the family $\pi_{\mathcal{M}}(X_i)_{k,\prec}$ follows the imposed ordering such that $X_j \prec X_i$. We can choose a graph $\mathcal{M}$ consistent with an order $\prec$ by choosing independently a family $\pi_{\mathcal{M}}(X_i)_{k,\prec}$ for each node. Therefore, summing over all possible graphs is equivalent to summing over all possible valid families. Considering that each of the k families has a score $\psi(X_i, \pi_{\mathcal{M}}(X_i)_{k,\prec}|\mathcal{D})$ and that the prior follows the modularity property we can rewrite Equation (3.26) (Friedman and Koller, 2003) as:

$$P(\mathcal{D}|\prec) \propto \sum_{\mathcal{M}\in\mathbb{M}} \prod_{i}^{N} \rho(X_i, \pi_{\mathcal{M}}(X_i))\psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D}) \tag{3.27}$$

$$= \prod_{i}^{N} \sum_{k} \rho(X_i, \pi(X_i)_{k,\prec})\psi(X_i, \pi(X_i)_{k,\prec}|\mathcal{D}) \tag{3.28}$$

This result asserts that we can sum over all the networks which are consistent with a given order, $\prec$, by summing the scores associated with each allowed family for each node and then multiplying them.

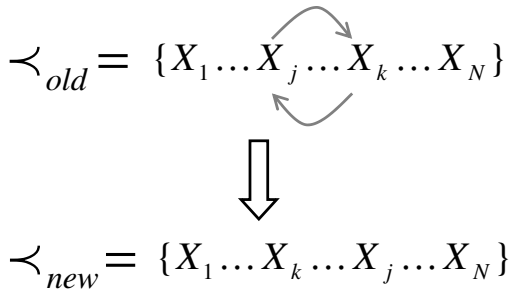


Figure 3.8: **Order MCMC proposal move.** The top figure presents the original order $\prec_{\text{old}}$ of N nodes. The bottom figure shows the order obtained by using the flip operation where the nodes X_j and X_k have their positions exchanged.

Now an MCMC approach is proposed as means to enable BNs consistent with all $n!$ possible orders over n nodes to be considered. Given an order $\prec_{\text{old}}$ a new order $\prec_{\text{new}}$ is proposed from the proposal distribution $Q(\prec_{\text{new}} \mid \prec_{\text{old}})$, which is then accepted according to the Metropolis et al. (1953); Hastings (1970) algorithm scheme with the following acceptance probability:

$$A(\prec_{\text{old}} \mid \prec_{\text{new}}) = \min\left\{\frac{P(\prec_{\text{new}} |\mathcal{D})Q(\prec_{\text{new}} \mid \prec_{\text{old}})}{P(\prec_{\text{old}} |\mathcal{D})Q(\prec_{\text{old}} \mid \prec_{\text{new}})}, 1\right\} \tag{3.29}$$

The proposal probability that is used in this thesis is a flip operation Friedman and Koller (2003). This consists in choosing two nodes in an order and then exchanging their positions in the order leaving all the other nodes unchanged. An example of such a proposal is presented in Figure 3.8. In this example the nodes X_j and X_k are exchanged in the original order, $\prec_{\text{old}}$ (top of figure), giving rise to the new order $\prec_{\text{new}}$ (bottom of figure).

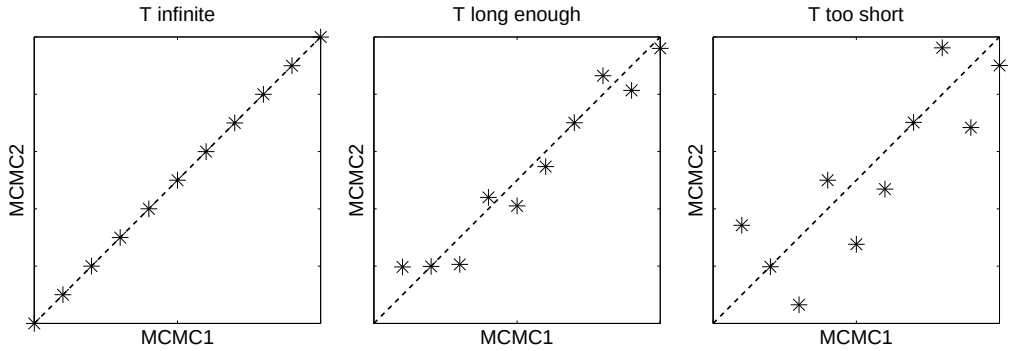
Whilst this proposal probability produces small steps on the space of orders it is very efficient to compute. For instance if we propose a move from $\prec_{\text{old}}$ to $\prec_{\text{new}}$ where nodes X_j and X_k are flipped those terms in Equation (3.28) which correspond to nodes that precede X_j or succeed X_k will not change. The only parent sets that need to have their scores recomputed are the ones that change according to the proposed order. This will include the parent sets for nodes between X_j and X_k which have either X_j or X_k in their composition. For such nodes the associated scores consistent with the actual ordering, $\prec_{\text{old}}$, have to be

subtracted and then the scores consistent with the new ordering, $\prec_{\text{new}}$, have to be added.

Order MCMC outputs a sample of node orders $\prec_1, \ldots, \prec_m$ which, if convergence of the Markov chain has been reached, is a sample from the posterior distribution over node orders $P(\prec |\mathcal{D})$. The idea of is to use this sample of orderings to obtain a sample of DAGs. To this end for each sampled ordering $\prec_i$ a DAG $\mathcal{M}_i$ is sampled out of the posterior distribution $P(\mathcal{M}| \prec, \mathcal{D})$. Thereby, as conditioned on the ordering, for each network node its parent set can be sampled independently with respect to its valid parent-sets in the ordering $\prec_i$. One of the known problems of the order MCMC is that if the prior over the orders $P(\mathcal{M}| \prec)$ is chosen to be uniform then the prior over the structures $P(\mathcal{M})$ will not be uniform. Graphs that are consistent with more orders are more likely. Several authors proposed corrections to order MCMC. One very recent and interesting approach is presented in Eaton and Murphy (2007).

3.2.2.4 Accessing MCMC convergence

One critical point when using MCMC samplers is to know whether or not the samples used for characterizing the distribution of interest are being sampled from the correct distribution. There are various tools for determining the convergence of MCMC samplers, see for example Cowles and Carlin (1996). In order to test the convergence of the MCMC simulations we resort to a simple heuristic approach. When sampling networks structures the result of the MCMC simulation is a matrix of posterior probabilities, P. Each entry p_{ij} of this matrix indicates the marginal posterior probability of the existence of an edge between nodes X_i and X_j. For accessing the MCMC's convergence we run the simulation twice from different initializations, obtaining two resulting posterior probability matrices, P^1 and P^2. We produce then a scatter plot by plotting p^1_{ij} against p^2_{ij}. As mentioned before this test is only a necessary but not a sufficient condition for



(a) (b) (c)

Figure 3.9: **Convergence test for MCMC simulations.** We plot the marginal posterior probabilities of the edges from two different simulation initializations. If the time t is infinite all the posterior probabilities of the edges for both simulations are going to be the same as exemplified in panel (a). If the time t is long enough we should expect a graph as in panel (b). If the simulations have not properly converged yet the graph should appear as in panel (c). Note that even if the graph looks like in panel (a), we only have information that the two simulations have reached the same meta-stable pre-equilibrium, which is a necessary condition but not a sufficient condition for MCMC convergence.

MCMC convergence.

3.2.3 Equivalence Classes of Bayesian Networks

Figure 3.10 shows four small distinct DAGs, which have the same skeleton but differ in their edge directions. We can expand the joint probability for each of these DAGs. Beginning from left to right, for the first DAG we have:

$$P(A, B, C) = P(C)P(A|C)P(B|C) \tag{3.30}$$

For the 2nd DAG:

$$\begin{aligned} P(A, B, C) &= P(A)P(C|A)P(B|C) \\ &= P(C)P(A|C)P(B|C) \end{aligned} \tag{3.31}$$

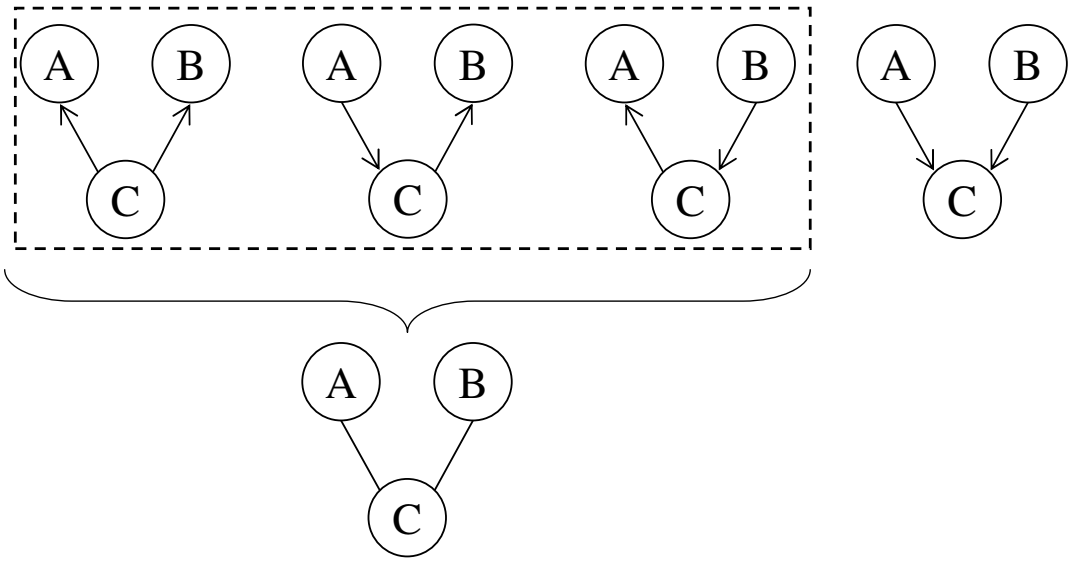


Figure 3.10: **Elementary BNs.** The top row shows four elementary BNs with the same skeleton but different edge directions. If we expand the joint probability to each one of these networks we can see that the first three, from left to right, are equivalent, being impossible to distinguish among them on the basis of the data. They are said to be equivalent networks and are represented by the CPDAG, which is presented in the bottom row.

For the 3th DAG:

$$\begin{aligned} P(A,B,C) &= P(A|C)P(C|B)P(B) \\ &= P(C)P(A|C)P(B|C) \end{aligned} \tag{3.32}$$

Finally for the 4th DAG:

$$P(A,B,C) = P(C|A,B)P(A)P(B) \tag{3.33}$$

The expansion of the joint probability for the first three DAGs shows that they are the same hence these three networks are said to be equivalent. Although the first three graphs are different they only represent alternative ways of explaining the same set of conditional independence relationships. Verma and Pearl (1991) proves that two DAGs are equivalent if and only if they have the same skeleton and the same set of v-structures. The skeleton of a DAG is the DAG with all edge directions removed, in other words, it is a DAG converted to a completely undirected graph. A v-structure is a configuration where two nodes are parents of

a third node and these two parents have no connection between them. According to Chickering (2002) a v-structure can be more precisely defined as:

- Given a set of n nodes.
- Select one triple of nodes (X_i, X_j, X_k) where $(i, j, k) \in \{1, \ldots, n\}$.
- This will be a v-structure if and only if:
    1. The DAG contains the directed edges $X_i \rightarrow X_j$ and $X_k \rightarrow X_j$.
    2. The DAG does not contain any edge connecting X_i and X_k

An equivalence class can be represented by a Partially Directed Acyclic Graph (PDAG), which is a graph containing both directed and undirected edges. In a PDAG every directed edge $X_i \longrightarrow X_j$ denotes that all the DAGs in this equivalence class contain the same edge. Conversely, the undirected edge $X_i - X_j$ means that some DAGs in this equivalence class contain the edge $X_i \longrightarrow X_j$ and some contains the edge in the opposite direction $X_i \longleftarrow X_j$. Equivalent BNs can not be distinguished on the basis of the data, the consequence is that all the edge directions that lead to equivalent classes must be discarded from the learned network. So if we infer one of the first three DAGs in the top row of Figure 3.10 what we really know in light of the data is a Partially Directed Acyclic Graph, PDAG, as represented in the bottom row of Figure 3.10.

The two scoring metrics that we use for scoring BNs (see Section 3.2.5) are score-equivalent. This means that they assert the same score for distinct networks that come from the same equivalence class. Because the result of our search algorithm is a DAG it is necessary to transform this DAG to the so called *completed* PDAG (CPDAG). This transformation from DAG to CPDAG is necessary since all the networks contained in the same equivalence class of the found DAG have the same score and, hence, they cannot be distinguished on the basis of their scores. In this thesis we use the algorithm presented by Chickering (2002) and

implemented by Grzegorczyk (2006) to efficiently obtain the CPDAG representation of the equivalence class from a given DAG.

3.2.4 Bayesian networks vs. causal networks

Although Bayesian networks are based on directed acyclic graphs (DAGs), it is important to note that not all directed edges in a Bayesian network can be interpreted causally. Like a Bayesian network, a causal network is mathematically represented by a DAG. However, the edges in a causal network have a stricter interpretation: the parents of a variable are its immediate causes. In the presentation of a causal network it is meaningful to make the causal Markov assumption: given the values of a variable's immediate causes, it is independent of its earlier causes. Under this assumption, a causal network can be interpreted as a Bayesian network in that it satisfies the corresponding Markov independencies. However, the reverse does not hold. The DAG on which the Bayesian network model is based just asserts a set of independence assumptions among the domain variables. More precisely, for each DAG we have that given a domain variable X and parent nodes $\pi_{\mathcal{M}}(X)$, X is independent of all its other ancestors. However, the same set of independence assumptions can often be asserted by different (equivalent) DAGs having the same skeleton but edges with opposite orientations, as discussed above in Section 3.2.3. Consequently, not every edge can indicate a causal relationship. The only way to interpret an edge causally is if we have no hidden variables and if all DAGs that are equivalent to each other (i.e. assert the same set of independence assumptions) agree on an edge direction, that is if the respective edge is directed in the corresponding CPDAG representation. In Section 3.4 we will discuss ways to increase the number of directed edges in equivalence classes by active interventions; in this way the number of putative causal interactions can be increased. However, a critical assumption made in this approach is the absence of any latent or hidden variables. If this assumption is

violated, the observation that two variables depend on each other probabilistically can be explained by the existence of an unobserved common cause. Since we are usually unable to rule out the existence of latent factors, we interpret the existence of directed edges in CPDAGs as *putative* causal interactions, which ultimately require an experimental validation. For a more detailed treatment of this subject, see Cooper and Glymour (1999); Pearl (2000).

3.2.5 Scoring metrics for Bayesian Networks

In Section 3.2.2 we discussed that the integral in Equation (3.7) is analytically tractable when the data is complete and the prior $P(\mathbf{q}|\mathcal{M})$ and the likelihood $P(\mathcal{D}|\mathbf{q},\mathcal{M})$ satisfy certain regularity conditions as discussed in Heckerman (1994, 1995). In the next two sections we will discuss these scoring metrics.

3.2.5.1 Discrete Multinomial Bayesian scoring metric

The Bayesian Dirichlet likelihood equivalent scoring metric is widely known as the BDe score. In this score each variable is assumed to be associated with a multinomial distribution. It can only deal with discretized variables. Although the discretization can lead to some loss of information the BDe score is very flexible as it is able to model non-linear relationships. If we assume that the variable is discretized into r levels[2] we can write:

$$P(X_i = k|\pi_{\mathcal{M}}(X_i) = j) = \theta_{ijk} \tag{3.34}$$

where θ_{ijk} is the probability that the domain variable X_i takes on its k-th value $k = 1,\ldots,r$ given the j-th parent configuration of $\pi_{\mathcal{M}}(X_i)$ $(j = 1,\ldots,r_i)$. Note that r_i is the possible number of parent configurations and it is defined by the cardinalities of $\pi_{\mathcal{M}}(X_i)$, $r_i = r^{|\pi_{\mathcal{M}}(X_i)|}$. Considering a data set $\mathcal{D}$ where N_{ijk} is

[2]We assume that each node has the same number of discretized values to keep the notation simple. However, note that it is possible to relax this assumption.

the number of observations in $\mathcal{D}$ in which variable X_i has the value k and the configuration of $\pi_{\mathcal{M}}(X_i)$ is j we can write the likelihood as:

$$P(\mathcal{D}|\theta, \mathcal{M}) = \prod_{i=1}^{n} \prod_{j=1}^{r_i} \prod_{k=1}^{r} \theta_{ijk}^{N_{ijk}} \tag{3.35}$$

We can rewrite Equation (3.7) as:

$$P(\mathcal{D}|\mathcal{M}) = \int \prod_{i=1}^{n} \prod_{j=1}^{r_i} \prod_{k=1}^{r} \theta_{ijk}^{N_{ijk}} P(\theta|\mathcal{M}) d\theta \tag{3.36}$$

The following assumptions are necessary:

- **Global parameter independence:** The parameters of each variable are independent of the parameters of the other variables.

$$P(\theta|\mathcal{M}) = \prod_{i=1}^{n} P(\theta_i|\mathcal{M}) \tag{3.37}$$

- **Local parameter independence:** The parameters corresponding to each instance of the parents of a variable are independent of each other.

$$P(\theta_i|\mathcal{M}) = \prod_{j=1}^{r_i} P(\theta_{ij}|\mathcal{M}) \tag{3.38}$$

- **Parameter modularity:** If we have two different networks $\mathcal{M}_1$ and $\mathcal{M}_2$ with positive prior probabilities, and if the node X_i has the same parents in both networks then for each configuration j of its parents:

$$P(\theta_{ij}|\mathcal{M}_1) = P(\theta_{ij}|\mathcal{M}_2) \tag{3.39}$$

- **Complete data:** There are no missing data or hidden variables.

With these assumptions we can rewrite the parameter prior:

$$P(\theta|\mathcal{M}) = \prod_{i=1}^{n} P(\theta_i|\pi_{\mathcal{M}}(X_i)) = \prod_{i=1}^{n} \prod_{j=1}^{r_i} P(\theta_{ij1}, \ldots, \theta_{ijr}) \tag{3.40}$$

Now Equation (3.36) is combined with (3.40) and by using the independence of terms the integral is rearranged, which yields:

$$P(\mathcal{D}|\mathcal{M}) = \prod_{i=1}^{n} \prod_{j=1}^{r_i} \int \left(\prod_{k=1}^{r} \theta_{ijk}^{N_{ijk}} \right) P(\theta_{ij1}, \ldots, \theta_{ijr}) d(\theta_{ij1}, \ldots, \theta_{ijr}) \tag{3.41}$$

The Dirichlet distribution is the conjugate prior to the Multinomial distribution and Heckerman et al. (1995) show that if we assume the prior distribution over parameters to be a Dirichlet we can find a closed form solution to Equation (3.41). The Dirichlet prior is given by:

$$P(\theta_{ij1}, \ldots, \theta_{ijr}) = \prod_{k=1}^{r} \theta_{ijk}^{\alpha_{ijk}-1} \left(\frac{\Gamma\left(\sum_{k=1}^{r} \alpha_{ijk}\right)}{\prod_{k=1}^{r} \Gamma\left(\alpha_{ijk}\right)} \right) \tag{3.42}$$

where the α_{ijk} are unknown hyperparameters and $\Gamma(.)$ is the Gamma function defined as:

$$\Gamma(x) \equiv \int_0^\infty u^{x-1} e^{-u} du, \quad x > 0. \tag{3.43}$$

Substituting the Dirichlet prior on Equation (3.41) Cooper and Herskovits (1992) show that the closed form solution to Equation (3.41) is given by:

$$P(\mathcal{D}|\mathcal{M}) = \prod_{i=1}^{n} \prod_{j=1}^{r_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij} + \alpha_{ij})} \prod_{k=1}^{r} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})} \tag{3.44}$$

where $N_{ij} = \sum_{k=1}^{r} N_{ijk}$ and $\alpha_{ij} = \sum_{k=1}^{r} \alpha_{ijk}$.

Buntine (1991) shows that the following choice of hyperparameters:

$$\alpha_{ijk} = \frac{\alpha}{r r_j} \tag{3.45}$$

with $\alpha > 0$ leads to score equivalence on BDe scoring scheme. As discussed in Section 3.2.3 equivalent networks are networks that although having different edges with different orientation cannot be distinguished in light of the data. See Figure 3.10 for examples of equivalent networks.

In our simulations using the BDe score we followed Buntine (1991) and always set $\alpha = 1$ which leads to a vague prior over the parameters since it produces relatively low hyperparameters α_{ijk}. The hyperparameters can be interpreted as the number of imaginary observations in which $X_i = k$ and $\pi_{\mathcal{M}}(X_i) = j$.

Following Equation (3.11) we can write the marginal likelihood of Equation (3.44) as a local score:

$$\psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D}) = \prod_{j=1}^{r_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij} + \alpha_{ij})} \prod_{k=1}^{r} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})} \tag{3.46}$$

where, again, $\psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D})$ is the score of the structure formed by node X_i and their parents $\pi_{\mathcal{M}}(X_i)$ given the data $\mathcal{D}$.

In summary, having observed some data $\mathcal{D}$ and having a model $\mathcal{M}$ we can calculate the marginal likelihood $P(\mathcal{D}|\mathcal{M})$ by integrating out the parameters according to Equation (3.44). These scores are referred as BDe scores.

3.2.5.2 Continuous Bayesian Gaussian scoring metric

The continuous Bayesian Gaussian likelihood equivalent scoring metric is known as the BGe score. Here we present this scoring metric as it is described by Geiger and Heckerman (1994). In this case each variable is assumed to be associated with a Gaussian distribution. Each variable X_i has a mean value $E[X_i]$ which depends on the values of its parent variables $\pi_{\mathcal{M}}(X_i) = (X_1, \ldots, X_j)$. The distribution of X_i is given by:

$$X_i \sim N\left(m_i + \sum_{j=1}^{n} b_{ij}(X_j - m_j), 1/v_i\right) \tag{3.47}$$

where m_i is the unconditional mean of X_i, $1/v_i$ is its conditional variance and b_{ij} coefficients represent the strength of the dependencies between X_i and X_j variables. Note that if $b_{ij} = 0$ there is no influence from variable X_j upon X_i. Conversely if $b_{ij} \neq 0$ there is an influence from X_j upon X_i. Hence, if $b_{ij} \neq 0,\ j < i$ we can conclude that the edge $X_j \rightarrow X_i$ does exist.

The precision matrix W of the joint multivariate Gaussian distribution over the n domain variables can be computed using the coefficients b_{ij} and the conditional variances $1/v_i$. In order to calculate W a recursive formula is applied. Assume that $W(i)$ denotes the $i \times i$ upper left submatrix of W, $\vec{b}_i$ denotes the column vector $(b_{1,i}, \ldots, b_{i-1,i})$ and $\vec{b}'_i$ denotes the transposed vector $\vec{b}_i$.

$$W(i+1) = \begin{pmatrix} W(i) + \frac{\vec{b}_{i+1}\vec{b}'_{i+1}}{v_{i+1}} & -\frac{\vec{b}_{i+1}}{v_{i+1}} \\ -\frac{\vec{b}'_{i+1}}{v_{i+1}} & \frac{1}{v_{i+1}} \end{pmatrix}$$

for $i > 0$ and $W(1) = \frac{1}{v_1}$.

Using this recursive formula $W(n)$ is the precision matrix for the joint Gaussian distribution of variables $X_1, \ldots, X_n$. The inverse of the precision matrix is the covariance matrix, $\Sigma = W^{-1}$. If we have defined the unconditional mean vector as $m = (m_1, \ldots, m_n)'$ then the joint Gaussian distribution is given by: $(X_1, \ldots, X_n) \sim N(m, \Sigma)$. Thus, it is shown above how a multivariate Gaussian distribution can be interpreted as a Gaussian belief network.

The main steps taken by Geiger and Heckerman (1994) which similarly to the BDe scores lead to a scoring scheme for Gaussian networks are presented here.

Assumption 1: the database $\mathcal{D}$ is a random sample from a multivariate Gaussian distribution with unknown means $\vec{m}$ and unknown precision matrix W.

Assumption 2: all the databases are complete. There are no missing data or hidden variables.

Assumption 3: the prior distribution over the unknown parameter $\vec{m}$ is a Gaussian distribution with mean vector $\vec{\mu_0}$ and precision matrix νW with $\nu > 0$. Furthermore, the matrix W is Whishart distributed with $\alpha > n + 1$ degrees of freedom and precision matrix T_0, denoted $w(\alpha, T_0)$ and defined as:

$$w(\alpha, T_0) = c(n, \alpha)|T_0|^{\frac{\alpha}{2}}|W|^{\frac{\alpha-n-1}{2}}e^{-\frac{1}{2}\mathrm{tr}(T_0 W)} \tag{3.48}$$

where $|.|$ and $\mathrm{tr}(.)$ are respectively the determinant and the trace of the matrix. The factors $c(n, \alpha)$ are given by:

$$c(n, \alpha) = \left(2^{\frac{\alpha n}{2}}\pi^{\frac{n(n-1)}{4}}\prod_{i=1}^{n}\Gamma\left(\frac{\alpha + 1 - i}{2}\right)\right)^{-1} \tag{3.49}$$

Given a random sample $\vec{x}_1, \ldots, \vec{x}_l$ it follows that the conditional distribution of $\vec{m}$ given W is a multivariate Gaussian distribution with mean vector $\vec{\mu}_n$ and precision matrix $(\nu + l)W$ where

$$\vec{\mu}_l = \frac{\nu\vec{\mu}_0 + l\bar{x}_l}{\nu + l} \qquad \bar{x}_l = \frac{1}{l}\sum_{i=1}^{l}\vec{X}_i \tag{3.50}$$

and the marginal of W is $w(\alpha + l, T_l)$, where T_l is given by:

$$T_l = T_0 + S_l + \frac{\nu l}{\nu + l}(\vec{\mu}_0 - \bar{x}_l)(\vec{\mu}_0 - \bar{x}_l)' \tag{3.51}$$

$$S_l = \sum_{i=1}^{l}(\vec{X}_i - \bar{x}_l)(\vec{X}_i - \bar{x}_l)' \tag{3.52}$$

where $\bar{x}_l$ is the sample mean and S_l is the sample variance of the database.

Note that ν, $\vec{\mu}_0$, α and T_0 are unknown hyperparameters that have to be specified in advance. We will discuss how to set these hyperparameters later. Afterwards Geiger and Heckerman (1994) show that the assumption of a normal-Wishart prior is sufficient for deriving the score for a complete Bayesian network denoted here by $\mathcal{M}^C$. By complete Bayesian network the authors designate networks with as many edges as possible, that is, $b_{ij} \neq 0$ for all $i < j$. The score for such networks is given by:

$$P(\mathcal{D}|\mathcal{M}^C) = (2\pi)^{-nm/2}\left(\frac{\nu}{\nu+m}\right)^{n/2}\frac{c(n,\alpha)}{c(n,\alpha+m)}|T_0|^{\frac{\alpha}{2}}|T_m|^{-\frac{\alpha+m}{2}} \tag{3.53}$$

With further two assumptions:

Parameter independence: Unknown parameters of the local probability distributions (Equation 3.47) are independent.

Parameter modularity: The prior distribution of the parameters of these local probability distributions depends on the parent variables only.

Geiger and Heckerman (1994) show that it is possible to derive a score for any DAG:

$$P(\mathcal{D}|\mathcal{M}) = \prod_{i=1}^{n}\frac{P(\mathcal{D}^{X_i\pi_{\mathcal{M}}(X_i)}|\mathcal{M}^C)}{P(\mathcal{D}^{\pi_{\mathcal{M}}(X_i)}|\mathcal{M}^C)} \tag{3.54}$$

where $\mathcal{D}^{\pi_{\mathcal{M}}(X_i)}$ and $\mathcal{D}^{X_i\pi_{\mathcal{M}}(X_i)}$ is the data set $\mathcal{D}$ restricted to the variables in $\pi_{\mathcal{M}}(X_i)$ and to the variables in $X_i \cup \pi_{\mathcal{M}}(X_i)$ respectively. This score is referred to as the BGe score.

Following Equation (3.11) we can write the marginal likelihood of Equation (3.54) as a local score:

$$\psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D}) = \frac{P(\mathcal{D}^{X_i\pi_{\mathcal{M}}(X_i)}|\mathcal{M}^C)}{P(\mathcal{D}^{\pi_{\mathcal{M}}(X_i)}|\mathcal{M}^C)} \tag{3.55}$$

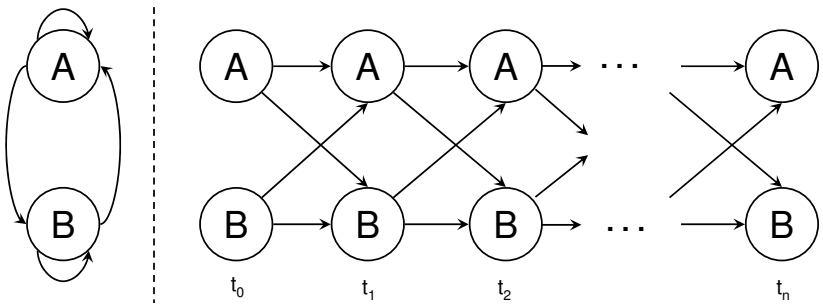


Figure 3.11: **Dynamic Bayesian Network.** The network to the left is not a proper DAG, the two genes interact with each other and have feedback loops. Considering delays between these interactions, it is possible to imagine this network *unfolded in time* where interactions within any time slice t are not permitted. The result is a proper DAG as the network represented on the right.

where, again, $\psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D})$ is the score of the structure formed by node X_i and their parents $\pi_{\mathcal{M}}(X_i)$ given the data $\mathcal{D}$.

Geiger and Heckerman (1994) discuss a heuristic method for defining T_0 and μ_0. The authors suggest that the user should specify a Gaussian network according to their knowledge. As an example suppose a Gaussian Bayesian network without any edge between the nodes and where each variable has a standard Gaussian distribution, $N(0, 1)$. It is possible to use the parameters that define such a network $(m_i, b_{ij}, 1/v_i)$ to obtain reasonable prior parameters:

$$\vec{\mu}_0 = \vec{m} \tag{3.56}$$

$$T_0 = \left(\frac{\nu(\alpha - n - 1)}{\nu + 1}\right)\Sigma \tag{3.57}$$

The parameter $\nu > 0$ is the equivalent sample size for $\vec{m}$ and the parameter $\alpha > n + 1$ is the equivalent sample size for matrix T_0. The higher these values are set, the more prior knowledge is implied through the prior network.

3.3 Dynamic Bayesian Networks

The previously mentioned BNs have some shortcomings. One important shortcoming is that it is impossible to model feedback loops, which are known to be present in real biological networks. Also when applying standard MCMC methods it is necessary to check the acyclicity of proposed structures; this checking of

acyclicity is one of the bottlenecks of MCMC simulations. One way to address these problems is to consider Dynamic Bayesian Networks (DBNs).

Consider the left structure in Figure 3.11, where two genes interact with each other via feedback loops. Note that this structure is not a valid Bayesian network as it violates the acyclicity constraint. When we unfold the network in the left panel of Figure 3.11 in time, as represented in the right panel of the same figure, we obtain a proper DAG and hence a valid BN again, the so-called Dynamic Bayesian Network (DBN). For more details about DBNs, see Friedman et al. (1998); Murphy and Milan (1999) and Husmeier (2003). We want to restrict the number of parameters to ensure they can be properly inferred from the data. For this reason, we model the dynamic process as a homogeneous Markov chain, where the transition probabilities between adjacent time slices are time-invariant. Intra-slice edges are not allowed since they would represent instantaneous 'time-less' interactions. Note that due to the direction of the arrow of time, the symmetry of equivalence classes is broken: the reversal of an edge would imply that an effect is preceding its cause, which is impossible. Summarizing, with DBNs we solve three shortcomings of static BNs: it is possible to model feedback loops, the acyclicity of the graph is automatically guaranteed by construction, and the symmetries within equivalence classes are broken, thereby removing any intrinsic ambiguities. Note, however, that the intrinsic assumption of DBNs is that the data have been generated from a homogeneous Markov chain, which may not hold in practice.

In practice when applying DBNs we only need to modify Equation 3.1 in order to incorporate the first order Markov assumption, which implies that a node $X_i(t)$ at time t has parents $\pi_{\mathcal{M}}(X_i)(t-1)$ at time $t-1$:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i(t)|X_{\pi_{\mathcal{M}}(X_i)}(t-1)) \tag{3.58}$$

where n is the total number of nodes. The application of DBNs with either BDe or BGe scores is straightforward.

3.4 Bayesian Networks with external interventions

Nowadays molecular biology has different techniques for producing interventions in biological systems, for instance, knocking genes down with RNA interference or transposon mutagenesis. The consequence is that the components of the system which are targeted by the interventions are no longer subject to the internal dynamics of the system under investigation. The components of the biological system can be either activated (up-regulated) or inhibited (down-regulated) and under this external intervention their values are no longer stochastic. The intervened components are not subject to the internal dynamics of the system, hence their values are deterministic. However, the other components which are not intervened are influenced by these deterministic values. Therefore, interventions are very useful to break the symmetries within the equivalence classes and consequently to the discovery of putative causal relationships. For a discussion about equivalence classes see Section 3.2.3 and for a discussion about putative causal relationships see Section 3.2.4.

In order to incorporate the interventions under the BN framework two small modifications are necessary.

First for observational data the likelihood $P(\mathcal{D}|\mathcal{M})$ as defined in Equation (3.11) is:

$$P(\mathcal{D}|\mathcal{M}) = \prod_{i}^{n} \psi(X_i, \pi_{\mathcal{M}}(X_i)|\mathcal{D})$$

and for a mixture of observational and interventional data this equation is modified to:

$$P(\mathcal{D}|\mathcal{M}) = \prod_{i}^{n} \psi(X_i, \pi_{\mathcal{M}}(X_i)|_{X_i \notin \mathcal{I}} \mathcal{D}) \tag{3.59}$$

where $\mathcal{I}$ is the set of interventions and $\underset{X_i \notin \mathcal{I}}{\mathcal{D}}$ denotes the data set where data points are removed for the cases where the node X_i is intervened with. Effectively Equation (3.59) says that the measurements of a node X_i under intervention are removed from the computation of the score.

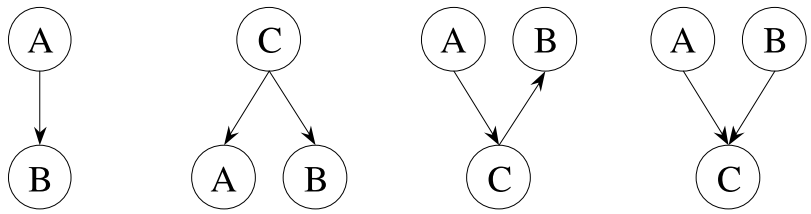


Figure 3.12: **Elementary interaction patterns.** *Left:* Direct interaction between two nodes. *Centre left:* Regulation of two nodes by a common regulator. *Centre right:* Signalling chain via an intermediate regulator. *Right:* Coregulation of a node by two regulators (v-structure).

The second necessary modification is related to the definition of equivalence classes. Tian and Pearl (2001) define the Transition Sequence equivalent networks (TS-equivalent). Two networks $\mathcal{M}_1$ and $\mathcal{M}_2$ are TS-equivalent if and only if they have the same skeleton, the same set of v-structures and the same set of parents for all manipulated variables. All edges connected with an intervened node become directed when the concept of TS-equivalence is applied. Therefore, new v-structures are formed and further edges become directed. In order to obtain the TS-equivalent DAG the procedure presented by Wernisch and Pournara (2004) is applied. For each intervened node in the network two dummy nodes are added each with one directed edge pointing from the dummy node to the intervened node. The new DAG now with the dummy nodes added is converted to a CPDAG (for a discussion about CPDAGs see Section 3.2.3). Finally the dummy nodes are removed and we have the DAG TS-equivalent graph.

3.5 Other Methods used to Infer Genetic Regulatory Networks

3.5.1 Relevance Networks

The method of relevance networks (RNs), proposed by Butte and Kohane (2000, 2003), is based on pairwise association scores. These scores are computed for all

pairs of nodes from the signals associated with the nodes. The authors propose the Mutual Information (MI) and the Pearson correlation as appropriate association scores.

The Pearson correlation coefficients are computed from continuous data and they can capture only relationships that are close to linear. If $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$ are the k-dimensional observations of variables x and y the Pearson correlation coefficient between these variables is given by:

$$\text{corr}(x, y) = \frac{\frac{1}{k}\sum_{i=1}^{k}(x_i - \overline{x})(y_i - \overline{y})}{\left(\sqrt{\frac{1}{k}\sum_{i=1}^{k}(x_i - \overline{x})^2}\right)\left(\sqrt{\frac{1}{k}\sum_{i=1}^{k}(y_i - \overline{y})^2}\right)} \tag{3.60}$$

where $\overline{x}$ and $\overline{y}$ are the empirical means of x and y respectively.

The MI scores are computed from discretized variables. The need of discretization can be seen as a disadvantage since it generally leads to some loss of information. On the other hand the MI scores can handle non-linear dependencies between the variables and this is an advantage over the scores which cannot handle non-linear dependencies. Considering that we have variables x and y discretized in r levels the MI between these variables can be defined as:

$$\text{MI}(x, y) = \sum_{i=1}^{r}\sum_{j=1}^{r} P(x = i, y = j) \log \frac{P(x = i, y = j)}{P(x = i)P(y = j)} \tag{3.61}$$

After having computed the scores (Pearson correlations or MI) for all possible pairs of variables in the domain some threshold is defined and the interactions that are above that threshold are preserved to compose the inferred network. Note that with either score the inferred network is intrinsically undirected due to the fact that corr(x, y) =corr(y, x) and MI(x, y) =MI(y, x).

The RN approach either with the Pearson correlation coefficients scores or with the MI scores is straightforward to implement, and its computational costs are comparatively low. The main disadvantage of RNs, however, is that the inference of an interaction between two nodes is not done in the context of the whole set of variables. Consequently, we do not expect RNs to be particularly powerful

in distinguishing between direct (Figure 3.12(a)) and indirect (Figure 3.12(b,c)) interactions.

3.5.2 Graphical Gaussian Models

Graphical Gaussian models (GGMs) are undirected probabilistic graphical models that allow the identification of conditional independence relations among the nodes under the assumption of a multivariate Gaussian distribution of the data. The inference of GGMs is based on a (stable) estimation of the covariance matrix of this distribution. The element C_{ik} of the covariance matrix $\mathbf{C}$ is related to the correlation coefficient between nodes X_i and X_k. A high correlation coefficient between two nodes may indicate a direct interaction (Figure 3.12(a)), an indirect interaction (Figure 3.12(c)), or a joint regulation by a common (possibly unknown) factor (Figure 3.12(b)). However, only the direct interactions are of interest to the construction of a regulatory network. The strengths of these direct interactions are measured by the partial correlation coefficient ρ_{ik}, which describes the correlation between nodes X_i and X_k conditional on all the other nodes in the network. From the theory of normal distributions it is known that the matrix $\boldsymbol{\rho}$ of partial correlation coefficients ρ_{ik} is related to the inverse of the covariance matrix $\mathbf{C}$, $\mathbf{C}^{-1}$ (with elements C_{ik}^{-1}) (Edwards, 2000):

$$\rho_{ik} \;=\; -\left(\frac{C_{ik}^{-1}}{\sqrt{C_{ii}^{-1}C_{kk}^{-1}}}\right) \tag{3.62}$$

To infer a GGM, one typically employs the following procedure. From the given data, the empirical covariance matrix is computed, inverted, and the partial correlations ρ_{ik} are computed from Equation (3.62). The distribution of $|\rho_{ik}|$ is inspected, and edges (i, k) corresponding to significantly small values of $|\rho_{ik}|$ are removed from the graph. The critical step in the application of this procedure is the stable estimation of the covariance matrix and its inverse. Schäfer and Strimmer (2005a) have extensively explored alternative regularization methods based on

bagging in order to estimate the covariance matrix. In a more recent study Schäfer and Strimmer (2005b) proposed a novel covariance matrix estimator regularized by a shrinkage approach which outperforms the previous methods based on bagging. Hence, we use the shrinkage estimator throughout this thesis.

In Schäfer and Strimmer (2005b) the authors present a novel regularized shrinkage covariance estimator which is based on the concept of shrinkage and exploits the Ledoit Wolf lemma (Ledoit and Wolf, 2004) for analytic calculation of the optimal shrinkage. This novel shrinkage estimator $\widehat{\mathbf{C}}$ for the covariance matrix $\mathbf{C}$ is guaranteed to be non-singular, so that it can be inverted to obtain a new estimator $\widehat{\boldsymbol{\rho}} = (\widehat{\mathbf{C}})^{-1}$ for the matrix $\boldsymbol{\rho}$. The new shrinkage estimator is based on the following theoretical idea. It is known that the unconstrained maximum likelihood estimator $\widehat{\mathbf{C}}_{\mathrm{ML}}$ for the covariance matrix $\mathbf{C}$ has a high variance if the number of variables exceeds the number of observations. On the other hand there are many other possible constrained estimators that have a certain bias but a lower variance. The shrinkage approach combines the maximum likelihood estimator with one of these constrained estimators $\widehat{\mathbf{C}}_T$ in a weighted average:

$$\widehat{\mathbf{C}} = (1 - \lambda) \cdot \widehat{\mathbf{C}}_{\mathrm{ML}} + \lambda \cdot \widehat{\mathbf{C}}_T, \tag{3.63}$$

where $\lambda \in [0, 1]$ denotes the shrinkage intensity. The authors show that this regularized estimator outperforms both single estimators $\widehat{\mathbf{C}}_{\mathrm{ML}}$ and $\widehat{\mathbf{C}}_T$ in terms of accuracy and statistical efficiency. Furthermore they show that the Ledoit Wolf lemma can be used to estimate the optimal shrinkage intensity $\lambda^\star$. The optimal shrinkage intensity is obtained in a data driven fashion by explicitly minimizing a risk function. The risk function is the expected loss and the expected loss in this case is the mean squared error (MSE). The authors present a variety of possible covariance matrix targets, $\widehat{\mathbf{C}}_T$. However, they recommend and use in their experiments the "diagonal, unequal variance" as the constrained estimator or target. This target is defined in Equation 3.64. One very interesting and useful property about the chosen target is that when combined with the unconstrained

maximum likelihood estimator according to Equation (3.63) the resulting shrunk covariance matrix is automatically positive definite. The elements of this specific target are given by:

$$\widehat{C}_{T_{ij}} = \begin{cases} \widehat{C}_{\mathrm{ML}_{ii}} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{3.64}$$

and for this target covariance matrix they show that the optimal estimated shrinkage estimator, $\widehat{\lambda}^{\star}$, is given by:

$$\widehat{\lambda}^{\star} = \frac{\sum_{i \neq j} \widehat{\mathrm{Var}}(\widehat{C}_{\mathrm{ML}_{ij}})}{\sum_{i \neq j} \widehat{C}^2_{\mathrm{ML}_{ij}}} \tag{3.65}$$

where $\sum_{i \neq j} \widehat{\mathrm{Var}}(\widehat{C}_{\mathrm{ML}_{ij}})$ and $\widehat{C}_{\mathrm{ML}_{ii}}$ are unbiased estimates obtained from the data.

Chapter 4

Benchmark data and evaluation criteria

4.1 Introduction

Despite the large amount of postgenomic data generated from new experiments very little is known about the biological structures which originate these data. The limited knowledge about the structures from which the data is generated makes the assessment of the method's performance very difficult. It is always a good practice to assess the methods both with real and simulated data. For the real data it is very difficult to have the knowledge about the true structure but this type of data is very important because ultimately one is interested in discovering structures from real data. The simulated data has the advantage that the true structure is fully known but the main disadvantage is that this data is often not very similar to the real data and the assessment of the performance based solely in simulated data may be biased.

In this chapter we present the real data from flow cytometry experiments that we use in most of our simulations. We also present different ways of generating both observational and interventional simulated data. This chapter concludes

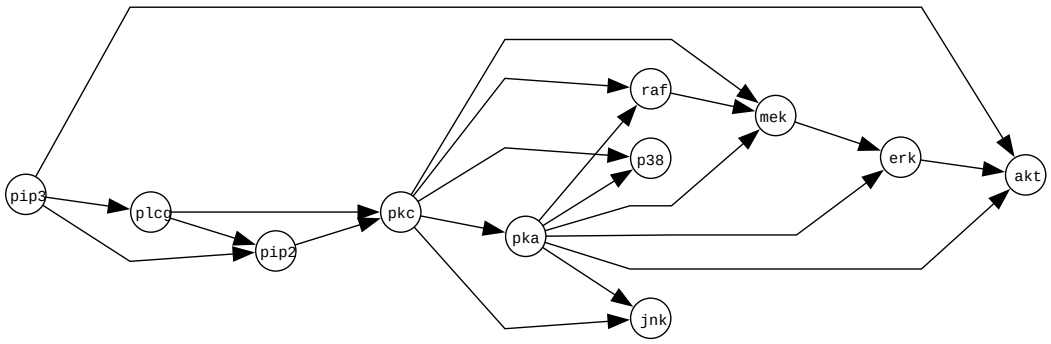


Figure 4.1: **Raf signalling pathway.** The graph shows the currently accepted signalling network, taken from Sachs et al. (2005). Nodes represent proteins, edges represent interactions, and arrows indicate the direction of signal transduction. In the interventional studies, the following nodes were targeted. Activations: PKA and PKC. Inhibitions: PIP2, AKT, PKC and MEK.

by presenting methods for assessing the performance of the reverse engineering methods.

4.2 Cytometry data

The Raf signalling network is depicted in Figure 4.1. Raf is a critical signalling protein involved in regulating cellular proliferation in human immune system cells. The deregulation of the Raf pathway can lead to carcinogenesis, and the pathway has therefore been extensively studied in the literature (e.g. Sachs et al. (2005); Dougherty et al. (2005)).

Sachs et al. (2005) have applied intracellular multicolour flow cytometry experiments to measure the expression levels of the 11 proteins that compose the network depicted in Figure 4.1. The proteins that had their expression measured are: `RAF`, `MEK`, `PLCg`, `PIP2`, `PIP3`, `ERK`, `AKT`, `PKA`, `PKC`, `P38` and `JNK`. Data were collected after a series of stimulatory cues and inhibitory interventions targeting specific proteins in the Raf pathway. The complete original data set is composed of 9 subsets, each related to one intervention. Table 4.1 presents the list of subsets and interventions that were applied to obtain the original data. Each of the subsets of data is composed of 600 measurements. In total we have

Data Set	Intervention
1	none
2	none
3	AKT inhibited
4	PKC inhibited
5	PIP2 inhibited
6	MEK inhibited
7	AKT activated
8	PKC activated
9	PKA activated

Table 4.1: **Original flow cytometry data set.** Table showing the interventions that are present in the original data set from Sachs et al. (2005). Each data set is composed by 600 samples. In total the original data set has 5400 data points from which 1200 are observational data and 4200 are interventional.

5400 data points from which 1200 are observational and 4200 are interventional. According to personal correspondence with the authors they have not used the subset of data where AKT was activated (number 7 in Table 4.1) therefore, in all our simulations we have also excluded this data set.

Flow cytometry allows the simultaneous measurement of the protein expression levels in thousands of individual cells. Sachs et al. (2005) have shown that for such a large data set, it is possible to reverse engineer a network that is very similar to the known gold standard Raf signalling network. However, for many other types of current postgenomic data, including microarray data, such abundance of data is not available. We therefore sampled the data of Sachs et al. (2005) down to 100 data points; this is a representative figure for the typical number of different experimental conditions in current microarray experiments. We averaged the results over 5 independent samples. We used the same sample size and the same number of replications for the synthetic data generation, which will be explained soon. The 5 observational data sets with 100 samples each were sampled from the the observational data. The observational data are the subsets of data which were

Data Points	Interventions
$1 \sim 16$	No Intervention
$17 \sim 30$	`AKT` inhibited
$31 \sim 44$	`PKC` inhibited
$45 \sim 58$	`PIP2` inhibited
$59 \sim 72$	`MEK` inhibited
$73 \sim 86$	`PKC` activated
$87 \sim 100$	`PKA` activated

Table 4.2: **Interventional data set.** Table showing how one interventional data set is built.

not intervened (subsets numbers 1 and 2 according to Table 4.1). Each of the 5 interventional data were obtained by sampling 16 unperturbed measurements and further 14 measurements for each of the 6 available interventions. Table 4.2 shows how each of the 5 intervened data sets is built.

The real flow cytometry data was preprocessed before being analysed. For interventions we occasionally observed a clear discrepancy between expected and observed concentrations for intervened nodes, e.g. some inhibitions hadn't led to low concentrations while some activations hadn't led to high concentrations. The missing changes in concentrations are not surprising, as most of the experimental interventions affected the activity of their targets instead of their concentrations. Correspondingly, for intervened nodes the measured concentrations do not reflect the strength of the true activity of the corresponding node (Karen Sachs, personal communication). Therefore, we decided to replace in each real interventional cytometric data set the values of the activated (inhibited) nodes by the maximal (minimal) concentration of that node measured under observational conditions. Afterwards, we used quantile-normalisation to normalise each real interventional data set. That is for each of the 11 variables (proteins) we replaced the 100 measured values by quantiles of the standard normal distribution $N(0, 1)$. More precisely, for each of the 11 variables (proteins) the j-th highest measured value was replaced by the $\left(\frac{j}{100}\right)$-quantile of the standard normal distribution, whereby

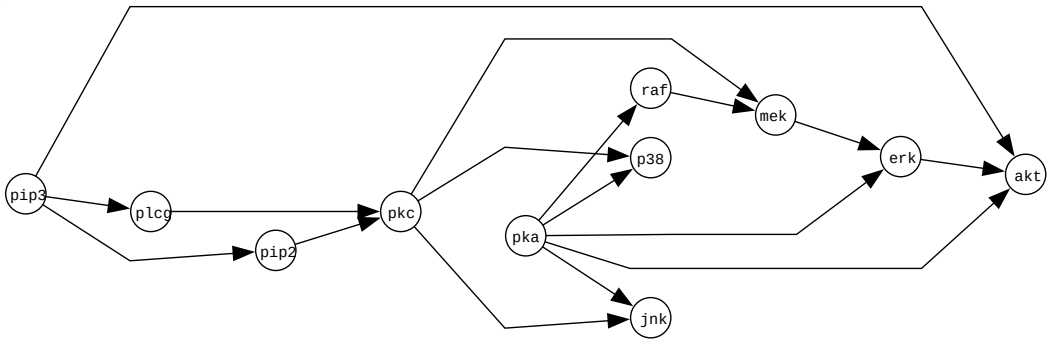


Figure 4.2: **Modified Raf signalling pathway.** The graph shows the modified Raf network where some of the edges were removed in order to increase the number of v-structures present in the network. From the original Raf network (Figure 4.1) the edges PKC → RAF, PKC → PKA, PKA → MEK and PLCg → PIP2 were removed, increasing the number of v-structures in the network.

the ranks of identical measured values were averaged.

4.2.1 The simulated v-structure network

When comparing different reverse engineering methods in Chapter 5 it is useful to have an idea about changes in the topology of the network. Therefore, we slightly modified the topology of the original network. With these modifications we have added v-structures to the network. Four edges were removed from the original topology and we have then 4 new v-structures in this new network. This so called v-structure network is presented in Figure 4.2. We generated synthetic data both from the original network and from the modified network, as discussed in the next section.

4.3 Simulating data from genetic regulatory networks

The main aim of this thesis is to investigate the performance and propose new algorithms for the learning of GRNs. Learning a GRN is in fact the learning of a network structure which indicates the dependence and independence relationships between variables that compose the network. The usual data used for this learn-

ing process are expression profiles from microarray experiments although other sources of data like protein expressions from flow cytometry experiments can also be used. The real biological expression data can come from different experimental settings. One of the differences is related to how the data is sampled; it can be either static or time series data. The static data denomination is used to refer to data in which one sample is collected after the biological system is submitted to some external challenge. The time series data denomination refers to data where after the biological system is challenged with some external condition a series of samples are collected, hence, we have a time ordering of the samples. Another difference about the nature of the collected data is related to whether the biological system is intervened or not. Observational data refers to data which are obtained from systems that are being passively 'observed', which means, they are not being externally modified. On the other hand interventional data are data in which some of the components of the system being examined are 'forced'. By 'forced' we mean that some of these components are either externally inhibited or externally activated, using e.g. gene knock-out experiments or over-expressions.

One problem is that in general the true network structure which generates the data is not known and therefore, it is very difficult to evaluate the performance of the learning algorithms. In order to be able to evaluate the algorithms' performance we resort to simulated data. The advantage of simulated data is that the network structure is known, making it possible to assess the learning algorithms' performance. Often simulated data are drawn from the multivariate Gaussian distribution while biological data rarely are Gaussian distributed. Another problem are the intricacies of the regulation by complex *cis*-regulatory modules, which makes the data far from being linearly dependent (Pournara, 2005). A model to simulate GRNs must be simple, possible to parametrize, and yet produce data that resemble biologically realistic data. With the advantages and shortcomings of the simulated data in mind we use two different ways of generating simu-

lated data. We generate data from a multivariate Gaussian distribution and we generate data using Netbuilder (Yuh et al., 1998, 2001), which is closer to real biological data. We explain both synthetic ways of generating observational and intervened data briefly in the next sections.

4.3.1 Gaussian simulated data

A simple synthetic way of generating data from a given structure is to sample them from a linear-Gaussian distribution. The random variable X_i (which denotes the expression of node X_i) is distributed according to

$$X_i \sim N\left(\sum_k w_{ik} X_k, \sigma^2\right) \tag{4.1}$$

where $N(.)$ denotes the Normal distribution, the sum extends over all parents of node X_i, and X_k represents both a node and the random variable associated with it. The interaction strength between nodes X_i and X_k is $w_{ik} \neq 0$. If $w_{ik} = 0$ then node X_k is not a parent of node X_i. The value of σ^2 can be interpreted as being the dynamic noise. Low values of σ^2 indicate a deterministic data set, that is, the value of the child node is almost completely determined by the value of its parents. Conversely high values of σ^2 indicate a very noisy data set. Given a network structure, in order to generate data with this method it is necessary to topologically sort the nodes first. This is necessary to guarantee that the parent nodes have their values computed before their child nodes.

From the linear Gaussian distribution we created 5 observational data sets and 5 interventional data sets. In Equation (4.1) we set the standard deviation to $\sigma = 0.1$, sampled the interaction strength $|w_{ik}|$ from the uniform distribution over the interval $[0.5, 2]$, and randomly varied the sign of w_{ik}. The interventional data sets are built in the same way as the interventional flow cytometry data sets, and Table 4.2 presents their composition. For simulating (noisy) interventions, we replace the conditional distribution (4.1) by the following unconditional distributions.

For inhibitions, we sample X_i from a zero-mean Gaussian distribution, $N(0, \sigma^2)$. For activations, we sample X_i from the tails of the empirical distribution of X_i, beyond the 2.5 and the 97.5 percentiles.

4.3.2 Netbuilder simulated data

A more realistic simulation of data typical of signals measured in molecular biology is the following approach. The expression of a gene is controlled by the interaction of various transcription factors, which may have an inhibitory or activating influence. Ignoring time delays inherent in transcription and translation, these interactions can be compared to enzyme-substrate reactions in organic chemistry. From chemical kinetics it is known that the concentrations of the molecules involved in these reactions can be described by a system of ordinary differential equations (ODEs) (Atkins, 1986). Assuming equilibrium and adopting a steady-state approximation, it is possible to derive a set of closed-form equations that describe the product concentrations as nonlinear (sigmoidal) functions of combinations of substrates. However, instead of solving the steady-state approximation to ODEs explicitly, as pursued in Pournara (2005), we approximate the solution with a qualitatively equivalent combination of multiplications and sums of sigmoidal transfer functions. The resulting sigma-pi formalism has been implemented in the software package Netbuilder (Yuh et al., 1998, 2001).

4.3.2.1 Enzyme substrate approximation

As mentioned above the sigma-pi formalism implemented in Netbuilder is based in enzyme-substrate reactions from organic chemistry. Here we present the main ideas of how to go from the enzyme-substrate reactions to the sigma-pi formalism. To model the dynamics of the processes inside the cell, it is necessary to remember some concepts from chemical kinetics. As an example, consider the first order

reaction where a reactant A is converted into a product B. It is represented by:

$$A \xrightarrow{k} B$$

The velocity, or rate of the reaction according with the law of mass action, u is given by (Atkins, 1986):

$$u = \frac{d[B]}{dt} = -\frac{d[A]}{dt} = k[A] \tag{4.2}$$

Applying the law of conservation of mass:

$$\begin{aligned} [A]_0 &= [A] + [B] \\ [A] &= [A]_0 - [B] \end{aligned} \tag{4.3}$$

$$u = \frac{d[B]}{dt} = -\frac{d[A]}{dt} = k[A] = k([A_0] - [B]) \tag{4.4}$$

where $[A]$ and $[B]$ are concentrations at a time t, $[A]_0$ is the initial concentration, and k is the rate constant. If we have a reversible reaction like,

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} B$$

where k_1 is the forward rate constant and k_{-1} is the backward rate constant, then the reaction rate is expressed as:

$$u = \frac{d[B]}{dt} = k_1[A] - k_{-1}[B] \tag{4.5}$$

Applying the law of conservation of mass:

$$\begin{aligned} [A]_0 &= [A] + [B] \\ [A] &= [A]_0 - [B] \\ \frac{d[B]}{dt} &= k_1([A]_0 - [B]) - k_{-1}[B] \\ &= k_1[A]_0 - [B](k_1 + k_{-1}) \end{aligned} \tag{4.6}$$

Knowing how to calculate the reaction's rate to first order reactions, a concept that was developed to describe small substrate-enzyme systems is used to

model the process of a protein (TF) binding to a TF binding site and starting transcription. Note that as this concept was developed for small molecules, and the systems that we are investigating are much larger (TF, TF binding sites and transcription), the transcriptional and translational time delays are being ignored.

Michaelis and Menten (1913) proposed a way to model how enzymes act as catalysers speeding up the conversion of a substrate into a product. They showed that as a concentration of a substrate increases, the rate of the reaction increases only up to a certain extent. They proposed the following mechanism for an enzyme substract reaction:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_2} E + P$$

where E is the enzyme, S is the substrate, P is the product and k_1 and k_2 are the association rates for the enzyme-substrate complex, ES, and the product respectively, and k_{-1} is the dissociation rate constant for the enzyme substract complex. Assuming that a steady-state will be reached, and then the concentration of ES will be constant, Briggs and Haldane (1925) derived the Michaelis and Mentem equation as follows:

$$\frac{d[ES]}{dt} = k1[E][S] - k_{-1}[ES] - k_2[ES] \tag{4.7}$$

Applying the law of conservation of mass:

$$[E]_0 = [ES] + [E] \tag{4.8}$$

$$[E] = [E]_0 - [ES]$$

then, assuming a steady-state is reached:

$$\frac{d[ES]}{dt} = k_1([E_0] - [ES])[S] - k_{-1}[ES] - k_2[ES] = 0 \tag{4.9}$$

and it follows that:

$$[ES] = \frac{k_1[E_0][S]}{k_{-1} + k_2 + k_1[S]} \tag{4.10}$$

so, the rate of reaction is given by:

$$u = k_2[ES] = \frac{k_2 k_1[E_0][S]}{k_{-1} + k_2 + k_1[S]} = \frac{k_2[E_0][S]}{\frac{k_{-1}+k_2}{k_1} + [S]} \tag{4.11}$$

Setting $V = k_2[E_0]$, where V is the limiting rate constant and $K_M = \frac{k_{-1}+k_2}{k_1}$, where K_M is the Michaelis constant, the equation can be rewritten as:

$$u = \frac{V[S]}{K_M + [S]} \tag{4.12}$$

The above derivations are for one enzyme and one substrate. When we use this concepts to model a TF and a TF binding site we should remember that TFs can act in various different forms to start transcription. Only as an example let us consider that there are three TFs: T_a, T_b and T_c. Some hypothetical possibilities are:

1. T_a alone gives rise to a transcription rate of a certain fraction of the maximum.

2. T_b alone doesn't initiate any transcription.

3. T_a and T_b together give rise to transcription at the maximum.

4. T_a and T_b and T_c repress the gene transcription.

In these few examples above we can see that the combined effect of different TFs is not necessarily the sum of their individual effects. We take an example from Pournara (2005) where two transcription factors are considered, one activating, T_a, and the other inhibiting, T_i, controlling the transcription of gene G. The system is described as:

$$T_i \underset{k_{13}}{\overset{k_{31}}{\rightleftharpoons}} G \underset{k_{21}}{\overset{k_{12}}{\rightleftharpoons}} T_a$$

Then the concentration of the mRNA of gene G is given by:

$$[G] = \frac{k_{12}[T_a]k_{31}}{k_{21}k_{31} + k_{12}[T_a]k_{31} + k_{13}[T_i]k_{21}} \tag{4.13}$$

Another possibility is the existence of more than one binding site, giving origin to what is called enzyme cooperativity. In this case the rate of reaction does not follow the Michaelis Menten Equation (4.12), but instead follows the equation proposed by Hill (1910), the so-called Hill equation, where a sigmoidal characteristic is evident:

$$u = \frac{V[S]^h}{K_{0.5}^h + [S]^h} \tag{4.14}$$

where $K_{0.5}^h$ is the Michaelis constant only when $h = 1$ and h is the Hill coefficient which gives an upper limit for the number of binding sites.

Summarizing, we started following a simple model of enzyme-substrate interaction (ignoring time delays). The use of chemical kinetics leads us to a set of ODEs describing the biophysical system. Assuming a steady state of this system, it is possible to derive a set of equations that describe the concentration of products as non-linear functions of combination of substrates. The resulting equations are a combination of multiplications and sums of sigmoidals. Thus instead of solving the steady-state approximation to ODEs explicitly, it is possible to model the system using the sigma-pi formalism, making the modelling much simpler with less parameters. This sigma-pi formalism has been implemented in the software package Netbuilder (Yuh et al., 1998, 2001). In the next section we explain how we simulate data using the Netbuilder software package.

4.3.2.2 Generating data with Netbuilder

The main idea of Netbuilder is instead of solving the steady-state approximation to ODEs explicitly we approximate them with a qualitatively equivalent combination of multiplications and sums of sigmoidal transfer functions. In Netbuilder pathways are represented as series of linked modules. Each module has specific input-output characteristics. As long as these characteristics conform to experimental observations, the exact transformations occurring inside the modules can

be safely neglected. The result is a significant reduction in the number of parameters. Thus, Netbuilder aims to provide a way of quantifying intuitively drawn diagrams, and making experimentalists hypotheses testable.

Netbuilder is a very flexible graphical tool to simulate biological systems. Here we explain the main components from Netbuilder that we have used in our data generation process. The principal interacting component in Netbuilder is called a "gene". The Netbuilder's gene has a user specified number of inputs and one output. The input(s) mimic the *cis*-regulatory domain where TF(s) can bind to initiate transcription and the output represents the protein concentration of the gene. The network topology is constructed by connecting the output of Netbuilder genes as inputs to other Netbuilder genes in the network. Genes that have no parents have their values sampled from the Uniform distribution over the interval $[0, 1]$.

If one gene has more than one parent Netbuilder offers different default continuous logical gates in order to combine the signal from the parents. In contrast with the classical logical gates, which are defined only for binary values, the continuous logical gates implemented in Netbuilder are defined for values on the interval $[0, 1]$. That means that the logical operations can assume any value on this interval. From now on when we refer to logical gates we are referring to the continuous logical gates implemented in Netbuilder. The default continuous logical gates `AND` and `OR` can be combined to produce other logical relationships as for example, `XOR`. Note that the gate `AND` mimics the situation where all the TFs are necessary to initiate transcription and the gate `OR` represents the situation where the presence of any TF is sufficient to the initiation of transcription.

In Netbuilder the value of the output of a gene is a deterministic function of its parents, but real biological systems are known to be noisy. Therefore, we add noise to Netbuilder genes in order to make the generated data more similar to real measured data. We add noise by using the sum function which is

implemented in Netbuilder and which permits a given value to be added to any other Netbuilder component. The value to be added is sampled from a normal distribution: $\epsilon \sim N(0, \sigma^2)$. In this manner we can control the level of noise by controlling the variance, σ^2.

Here we present some small examples in order to clarify how Netbuilder works in practice. Suppose we have a gene with only one parent. Let us assume that the parent value (concentration) is $[x]$ the value of its output, $[y]$, will then be:

$$[y] = \frac{[x]}{[x]+1} + \epsilon \tag{4.15}$$

Now let us consider that the gene has two parents with concentrations $[x_1]$ and $[x_2]$, which are combined through an `AND` gate. The output in this case is:

$$[y] = \left(\frac{[x_1]}{[x_1]+1}\right)\left(\frac{[x_2]}{[x_2]+1}\right) + \epsilon \tag{4.16}$$

Considering the same situation as before but now with the gate `OR` we have the output as:

$$[y] = \frac{[x_1]}{[x_1]+1} + \left[\left(\frac{[x_2]}{[x_2]+1}\right)\left(1 - \frac{[x_1]}{[x_1]+1}\right)\right] + \epsilon \tag{4.17}$$

Note that when using Netbuilder it is not necessary to produce the equations as presented here in the examples. Netbuilder is completely graphical and the equations are implicitly built through the selected components of the network. Although we presented situations with at the maximum two parents Netbuilder allows the user to define more than two. In all our simulations we used `OR` gates as the relationship between the parents and we used at maximum three parents.

In order to simulate the Raf network with Netbuilder we linked the 11 genes with the same structure as we have in the network presented in Sachs et al. (2005), see Figure 4.1. All the links between genes represent activations and all the interactions between TFs were set to `OR` regulation. A gene with an `OR` port is highly expressed if any of the TFs present in the input are high. For data

generation we sampled values from a uniform distribution, Uniform $\sim (0, 1)$, for all root nodes. Root nodes are nodes without any parents. These values are then propagated to the child nodes where they will be processed and then propagated further down in the network hierarchy until they reach the leaf nodes. Leaf nodes are nodes without any children. Every node that is not a root node has a sum function added to its output. This sum represents that the output of a node is subject to some additive noise. The values to be added as noise are sampled from a normal distribution $N \sim (0, \sigma^2)$. Using this method for adding dynamical noise we then generated data sets with three different noise levels: low, medium and high; corresponding to the following standard deviations: $\sigma = 0.01$, $\sigma = 0.1$ and $\sigma = 0.3$. Following this procedure we generated 5 observational data sets (with 100 data points each) for each noise level; these are called the Netbuilder observational data.

In order to generate interventional data with Netbuilder we proceed as follows. When an inhibition is simulated, the inhibited gene has its output forced to be zero independent of its inputs. After being forced to be zero the noise is added, thus the output of such an inhibited node will be only the added noise. In the case where we want to activate a gene we set its value to one, again independent of its input. The output of an activated gene will be 1 plus the noise. The noise added to the nodes subject to inhibitions or activations was always sampled from $N \sim (0, \sigma^2)$ with $\sigma = 0.01$.

We generated 5 interventional data sets for each noise level. These are called the Netbuilder interventional data. Each interventional data set is composed of a total of 100 data points where some of the genes were intervened; see Table 4.1 for a detailed explanation of how the interventional data set is built. As in the linear Gaussian data our interventions try to mimic the ones that were used in Sachs et al. (2005).

Real data	
Obs	Int

Gaussian data			
Original structure		v-structure	
Obs	Int	Obs	Int

Netbuilder data											
σ=0.01				σ=0.1				σ=0.3			
Original structure		v-structure		Original structure		v-structure		Original structure		v-structure	
Obs	Int	Obs	Int	Obs	Int	Obs	Int	Obs	Int	Obs	Int

Figure 4.3: **Data summary.** This figure presents a summary of all the data sets available. The abbreviations 'Obs' and 'Int' mean respectively data sets that are purely observational and data sets that are interventional. Each 'Obs' and 'Int' is composed by 5 data sets with 100 data points each. See the main text and the Table 4.1 for an explanation about how the interventional data sets are built. For the Gaussian data sets we always set $\sigma = 0.1$.

4.4 Evaluation metrics

All the methods evaluated in this thesis to reverse engineer networks produce as a result a matrix of scores associated with edges in the network. If we have n nodes in a network, the resulting matrix of scores $\mathcal{S}$ has dimension $n \times n$ and each entry $s_{ij} \in [0, 1]$ represents the score which indicates the strength of the relationship between nodes X_i and X_j. These scores are of different nature: correlation coefficients for RNs, partial correlation coefficients for GGMs, and marginal posterior probabilities for BNs.

As a means to assess the algorithms' performance it is necessary to compare it with some known network. We call this known network the true network $\mathcal{T}$ where the entries $t_{ij} \in \{0, 1\}$ indicate the presence and the absence of the connection between nodes X_i and X_j. In order to compare our resulting network $\mathcal{S}$ with the true network $\mathcal{T}$, we transform it to an adjacency matrix, $\mathcal{A}(\epsilon)$, by imposing a threshold ϵ. Each entry of the adjacency matrix is defined by:

$$a_{ij} = \begin{cases} 1, \ s_{ij} \geq \epsilon \\ 0, \ s_{ij} < \epsilon \end{cases} \tag{4.18}$$

Having these two matrices, $\mathcal{T}$ and $\mathcal{A}(\epsilon)$, we can classify each of the edges into categories. An edge can be classified as: true positive (TP), false positive (FP),

t_{ij}	a_{ij}	Category
0	0	TN
0	1	FP
1	0	FN
1	1	TP

Table 4.3: **Classification of edges.** This table shows how an edge is classified according to the values in the true matrix (t_{ij}) and in the adjacency matrix (a_{ij}). An entry that is equal to zero means that the edge from node X_i to node X_j is absent, conversely an entry that is equal to one means that the edge is present.

true negative (TN) or false negative (FN). TP is an edge which is present in $\mathcal{A}(\epsilon)$ and in $\mathcal{T}$. FP is an edge which is present in $\mathcal{A}(\epsilon)$ but is absent in $\mathcal{T}$. TN is a non-edge which is present in $\mathcal{A}(\epsilon)$ and in $\mathcal{T}$. FN is a non-edge which is present in $\mathcal{A}(\epsilon)$ but is absent in $\mathcal{T}$. Table 4.3 shows a summary of how the edges are classified into these categories.

The algorithms that we use for inferring the networks can result in graphs that are directed, undirected or partially directed. In order to assess the performance of these methods we apply two criteria. The first approach, referred to as the *undirected graph evaluation* (UGE), discards the information about the edge directions altogether. To this end, the original and learned networks are replaced by their skeletons, where the skeleton is defined as the network in which two nodes are connected by an undirected edge whenever they are connected by any type of edge. The second approach, referred to as the *directed graph evaluation* (DGE), compares the predicted network with the original directed graph. A predicted undirected edge is interpreted as the superposition of two directed edges, pointing in opposite directions. Figure 4.4 presents an example how the TPs and FPs are counted according to the UGE criteria and Figure 4.5 shows the equivalent example for the DGE criteria.

As already discussed the application of the algorithms to learning network

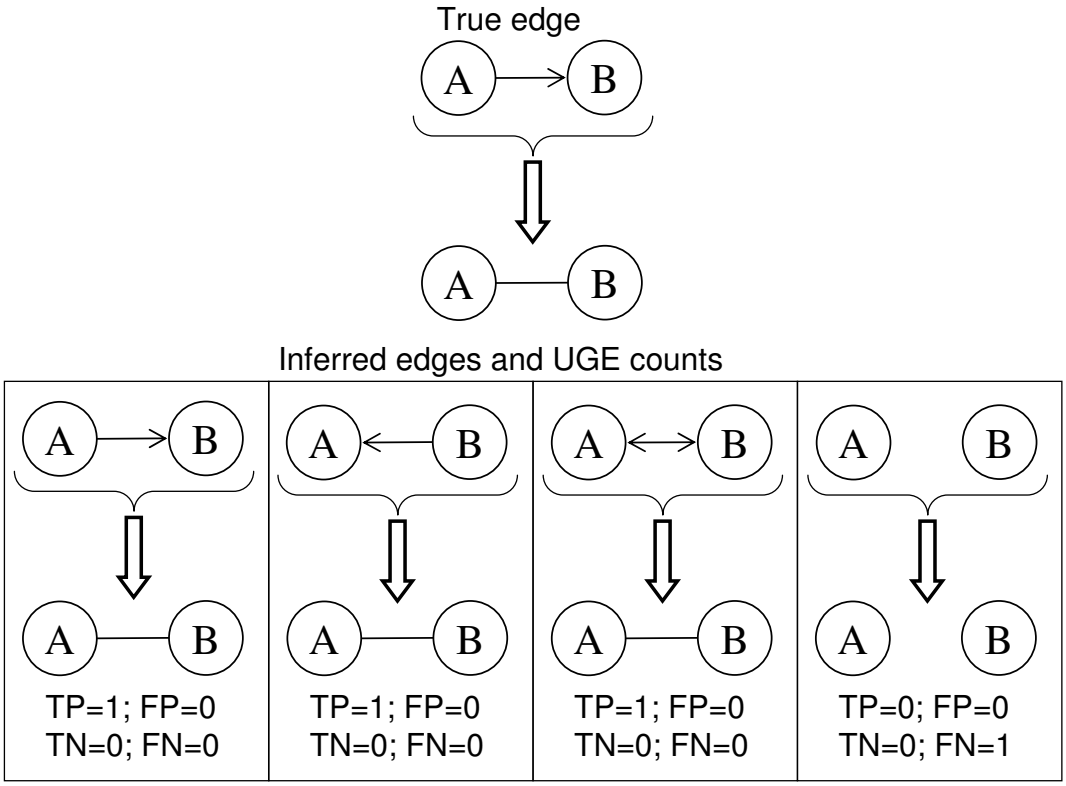


Figure 4.4: **UGE scoring.** This schematic example shows how the *undirected graph evaluation* UGE scoring criteria works. The top of the figure presents the true edge which according to this criterion is transformed into the undirected edge presented immediately below. The bottom of the figure shows the possible inferred directed edges and how they are transformed into undirected edges. For each of the potentially inferred edges the true positives (TP) and false positive (FP) counts are presented.

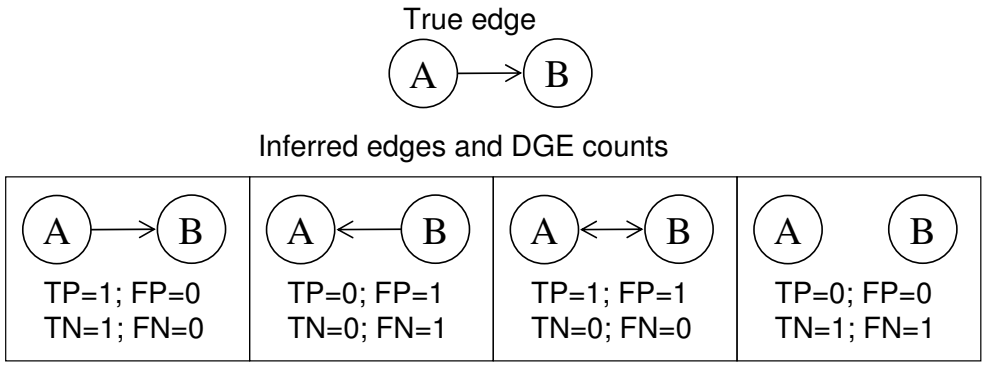


Figure 4.5: **DGE Scoring.** This schematic example shows how the *directed graph evaluation* DGE scoring criteria works. The top of the figure shows the true edge. The bottom of the figure shows the potentially inferred edges and their respective true positives (TP) and false positive (FP) counts.

structures leads to a matrix of scores $\mathcal{S}$ which defines a ranking of the edges. Given a threshold ϵ we can obtain an adjacency matrix $\mathcal{A}(\epsilon)$ which enable us to count the number of TPs, FPs, TNs and FNs. The receiver operator characteristics (ROC) curve is obtained by varying the threshold, ϵ, and plotting the relative number of TP edges against the relative number of FP edges for each of the thresholds.

The relative number of TP edges or sensitivity is defined by:

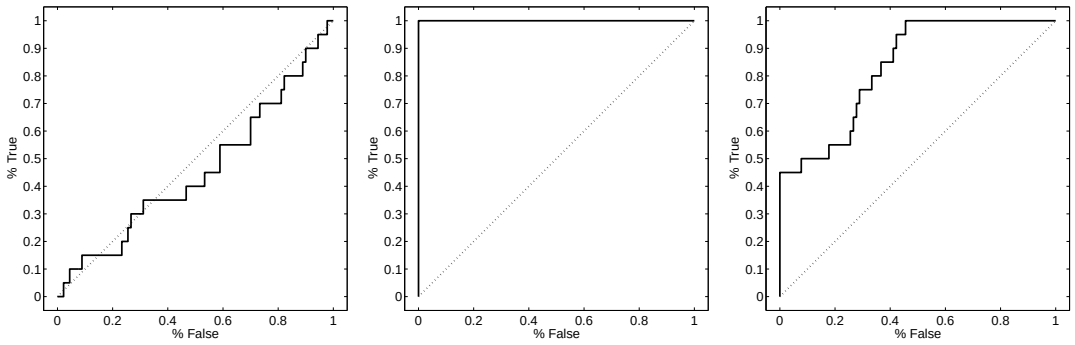
$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{4.19}$$

The relative number of FP edges or 1−specificity is defined by:

$$\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{4.20}$$

where the symbols TP, FP, TN and FN represent respectively the counts of the number of TPs, FPs, TNs and FNs.

Ideally we would like to evaluate the methods on the basis of the whole ROC curves. When too many results are being compared, unfortunately, this approach would not allow us to concisely summarize the findings. Hence, when there are many methods to compare we use the area under the ROC curve (AUC) instead. The AUC is a measure of the area under the ROC curve and summarizes the results for all the thresholds. In general bigger area values represent better predictors. However, this is not always the case. Note that in practice one is interested in low FP rates and therefore a curve with a rapid increase of the TP rate at the left part can be more advantageous than one with very low increase even if the total area of the first is lower than the total are of the second. Figure 4.6 present some ROC curve examples. Figure 4.6(a) shows the situation of a random predictor; in this case the AUC has a value around 0.5. Figure 4.6(b) shows the extreme case where the predictor is perfect; in this case all the TPs are recovered without any FP prediction, the respective AUC value is 1 in this



(a) (b) (c)

Figure 4.6: **ROC curve examples.** Here we show ROC curves examples. The left panel show a completely random predictor. The central panel shows the example of a perfect predictor, in this case all TPs are recovered without any FP and AUC$= 1$. The right panel shows a realistic example where some FPs are recovered for a certain number of recovered TPs.

case. The last example in Figure 4.6(c) is more realistic; it shows the case where a certain numbers of TPs are recovered for a smaller number of FPs.

While the AUC value does not require the commitment to the adoption of any (arbitrary) decision criterion, it does not lead to a specific network prediction. It also ignores the fact that, in practice, one is particularly interested in the performance for low FP rates. To address this shortcoming we define a second performance criterion based on the selection of a particular threshold on the edge scores, from which a specific network prediction is obtained. We fix the threshold such that it leads to FP=5. The evaluation is based on the TP counts we obtain when having this threshold fixed. This guarantees that we compare the methods at the same operation point on the ROC curve. Note that for different network sizes one may want to choose a different value for the fixed FP. As we are always using the same network size we kept this value at 5.

Chapter 5

Comparative evaluation of reverse engineering methods

This chapter describes the results of a collaboration study with Marco Grzegorczyk and Dirk Husmeier, published in Werhli et al. (2006).

Traditional approaches to systems biology are based on a mathematical description of putative pathways in terms of coupled differential equations with the objective to obtain a deeper understanding of the exact nature of the regulatory circuits and their regulation mechanisms. However, the availability of high-throughput postgenomic data has recently prompted substantial interest in reverse engineering the networks and pathways in an inferential way from the data themselves. One of the first seminal papers promoting this approach aimed to learn gene regulatory networks in *Saccharomyces cerevisiae* from gene expression profiles with Bayesian networks (Friedman et al., 2000). Since then, several authors have applied Bayesian networks to infer regulatory networks from postgenomic data of different nature (for instance, Imoto et al. (2003a); Nariai et al. (2005)). Various alternative methods, like relevance networks (Butte and Kohane, 2003) (see Section 3.5.1) and graphical Gaussian mod-

els (Schäfer and Strimmer, 2005a) (see Section 3.5.2) have been proposed and applied to the inference of gene regulatory networks from gene expression data. These methods are of enormous importance in the emerging field of genetical genomics (Bing and Hoeschele, 2005), where QTL marker analysis is first applied to identify putative sets of regulatory genes, from which then a more refined regulatory network is to be reverse engineered.

One of the first major evaluation studies was carried out by Smith et al. (2002). The authors simulated a complex biological system at different levels of organization, involving behaviour, neural anatomy, and gene expression of songbirds. They then tried to infer the structure of the known true genetic network from the simulated gene expression data with Bayesian networks. In a related study, Husmeier (2003) evaluated the accuracy of reverse engineering gene regulatory networks with Bayesian networks from data simulated from realistic molecular biological pathways, where the latter were modelled with a system of coupled differential equations. This network was also used in an earlier study by Zak et al. (2001), who investigated the inference accuracy of deterministic linear and log-linear models. While all three papers shed some light on the accuracy of reconstructing regulatory networks, they only investigated a particular inference method and do not include a cross-method comparison.

In order to address this shortcoming, an extensive evaluation study was carried out by Pournara, 2005. The author compared graphical Gaussian models and Bayesian networks on synthetic data generated from networks with random structures and different gene regulation mechanisms, where the latter differed with respect to the cooperative or competitive interactions between transcription factors regulating the same gene. The approach we present in this chapter is motivated by Pournara (2005) and complements this work in four important respects. First, the learning algorithm for Bayesian networks has been improved. In order to capture the uncertainty inherent in learning from sparse and noisy

data, we sample network structures from the posterior distribution with MCMC. This approach is methodologically more consistent than the optimization scheme applied in Pournara (2005). For the practical realization, we apply a sampling strategy based on node orders (Friedman and Koller, 2003) (for details see Section 3.2.2.3), which achieves faster mixing and convergence than conventional sampling in the space of network structures (Madigan and York, 1995). Second, we use improved inference methods for graphical Gaussian models. The approach adopted in Pournara (2005) is based on the PC algorithm of Spirtes et al. (2001). In the present work, we apply a more recent algorithm proposed by Schäfer and Strimmer (2005b), which the authors have developed after extensive experimentation with methods for stabilizing covariance matrix estimations (Schäfer and Strimmer, 2005a). For more details about Graphical Gaussian models see Section 3.5.2. Third, we include another reverse engineering method in our comparison: the approach of relevance networks proposed by Butte and Kohane (2000, 2003), for details see Section 3.5.1. This approach is appealing due to its low computational costs, and we investigate to what extent the results can be improved with the more complex alternative algorithms mentioned above. Fourth, rather than evaluating the performance on randomly generated network structures, we base our comparison on the Raf pathway, a critical protein signalling network involved in regulating cellular proliferation in human immune system cells (Sachs et al., 2005). Our evaluation exploits four types of data, distinguishing between passive observations and active interventions, and using data from both laboratory experiments as well as synthetic simulations.

5.1 Methods

In this study we compare three methods:

- Relevance networks (RNs), for details see Section 3.5.1.

- Graphical Gaussian models (GGMs), for details see Section 3.5.2.

- Bayesian Networks (BNs), for details see Section 3.2.

5.1.1 Observational versus interventional data

Modern molecular biology possesses an extensive inventory of techniques for targeted interventions, for instance, knocking genes down with RNA interference or transposon mutagenesis. The consequence is that targeted nodes are no longer subject to the internal dynamics of the system under investigation, and the respective terms have to be excluded from the expansion in Equation (3.1) in Section 3.2. This may break the symmetries within the equivalence classes; while equivalent structures have equal posterior probabilities under passive observations, this may no longer holds when subjecting the system to external interventions. Consequently, edge directions that are ambiguous under passive observations can be retrieved, and this forms the basis for learning putative causal interactions; see Section 3.4 for further details.

5.1.2 Comparison between the methods

GGMs and BNs potentially distinguish between direct and indirect interactions and therefore provide a more powerful modelling approach than RNs. BNs have the potential to present a more refined picture of interactions among nodes than GGMs due to the directed nature of the edges. For instance, the graph on Figure 3.12(d) represents a marginal independence of the parental nodes, A and B. However, conditional on measurements obtained for the child, node C, the parental nodes are dependent. The equivalent undirected graph contains an extra edge between the parents, and this so-called moralization (Heckerman, 1999) deteriorates the resolution of the independence relations. In addition, directed edges present putative indications of causal interactions (Heckerman, 1999) and provide

a straightforward model for accommodating interventional data. Finally, the inference procedure we adopt for learning BNs is score-based and more complex than the constraint-based approach adopted for GGMs (see Pournara (2005) for a comprehensive exposition of the difference between these two learning paradigms). The latter approach aims to 'explain away' an observed correlation between two nodes by testing whether this correlation is not the effect of a regulation by other nodes. To this end, the partial correlations are computed, that is, the correlations conditional on all the other nodes in the system. This approach does not take into account whether network configurations that explain away these correlations are truly present. The score-based approach is in principle more powerful in that it marginalizes over all possible network configurations. However, the respective integral is analytically intractable, and the numerical approximation with MCMC is computationally expensive. In fact, the robust estimation of a rank-deficient covariance matrix proposed by Schäfer and Strimmer (2005b) turns constraint-based inference with GGMs into an extremely fast and attractive approach. Hence, the objective of the present study is to investigate whether the application of the more complex score-based approach to learning BNs is of any practical benefit for reverse engineering gene regulatory networks.

5.2 Data

We base the evaluation of the three reverse engineering methods (RNs, GGMs and BNs) on the Raf signalling network, depicted in Figure 4.1. We use four types of data for our evaluation. First, we distinguish between passive observations and active interventions. Second, we use both real laboratory data as well as synthetic simulations. This combination of data is based on the following rationale. For simulated data, the true structure of the regulatory network is known; this allows us, in principle, to faithfully evaluate the prediction results. However, the model

used for data-generation is a simplification of real molecular-biological processes, and this might lead to systematic deviations and a biased evaluation. The latter shortcoming is addressed by using real laboratory data. In this case, however, we ultimately do not know the true signalling network; the current gold-standard might be disputed in light of future experimental findings. By combining both approaches, we are likely to obtain a more reliable picture of the performance of the competing methods. For a detailed description of all the data sets see Chapter 4.

5.3 Simulations

As opposed to GGMs, RNs and BNs do not require the assumption of a Gaussian distribution. However, deviations from the Gaussian incur an information loss as a consequence of data discretization (mutual information for RNs, BDe score for BNs). Alternatively, when avoiding the discretization with the heteroscedastic regression approach of Imoto et al. (2003b), the integral in Equation (3.7) becomes intractable and has to be approximated (using, e.g., the Laplace method). It would obviously be interesting to evaluate the merits and shortcomings of these nonlinear approaches. However, the main objective of the present study is the comparison of three modelling paradigms: (1) pairwise association scores independent of all other nodes (RNs), (2) undirected graphical models with constraint-based inference (GGMs), and (3) directed graphical models with score-based inference (BNs). To avoid the perturbing influence of additional decision factors, e.g. related to data discretization, and to enable a fair comparison with GGMs, we use the Gaussian assumption throughout.

Applying the Gaussian assumption to BNs, with the normal-Wishart distribution as a conjugate prior on the parameters, the integral in Equation (3.7) has a closed-form solution, referred to as the BGe score. For details see Section 3.2.5.2

and Geiger and Heckerman (1994). The score depends on various hyperparameters, which can be interpreted as pseudocounts from a prior network. To make the prior probability over parameters – $P(\mathbf{q}|\mathcal{M})$ in Equation (3.7) – as uninformative as possible, we set the prior network to a completely unconnected graph with an equivalent sample size as small as possible subject to the constraint that the covariance matrix is non-singular. For the prior over network structures – $P(\mathcal{M})$ in Equation (3.6) – we followed Friedman and Koller (2003) and chose a distribution that is uniform over parent cardinalities (see Section 3.2.2) subject to a fan-in restriction of 3. We carried out MCMC over node orders, as proposed in Friedman and Koller (2003) and presented in Section 3.2.2.3. To test for convergence, each MCMC run was repeated from two independent initializations. Consistency in the marginal posterior probabilities of the edges was taken as indication of sufficient convergence. This method for testing convergence is explained in more detail in Section 3.2.2.4. We found that a burn-in period of 20,000 steps was usually sufficient, and followed this up with a sampling period of 80,000 steps, keeping samples in intervals of 200 MCMC steps. For RNs, we computed the pairwise node associations with the Pearson correlation, see Section 3.5.1. We computed the covariance matrix in GGMs with the shrinkage approach proposed by Schäfer and Strimmer (2005b), choosing a diagonal matrix as the shrinkage target; for more details see Section 3.5.2. Note that this target corresponds to the empty prior network; hence the effect of shrinkage is equivalent to the selected prior for the computation of the BGe score in BNs. The practical computations were carried out with the software provided by Schäfer and Strimmer (2005b). The MCMC simulations were carried out with our own MATLAB© programs.

5.4 Evaluation

While the true network is a directed graph, our reconstruction methods may lead to undirected, directed, or partially directed graphs. To assess the performance of these methods, we apply two different criteria, namely UGE and DGE evaluation. These two criteria were explained in detail in Section 4.4. Each of the three reverse engineering methods compared in our study leads to a matrix of scores associated with the edges in a network. These scores are of different nature: correlation coefficients for RNs, partial correlation coefficients for GGMs, and marginal posterior probabilities for BNs. However, all three scores define a ranking of the edges. This ranking defines a receiver operator characteristics (ROC) curve, where the relative number of true positive (TP) edges is plotted against the relative number of false positive (FP) edges. The ROC curve is explained in more detail in Section 4.4.

In order to evaluate our results we use two evaluation metrics: the area under the ROC curve (AUC) and the number of recovered true positives (TP) for a fixed number of 5 false positives (FP). These two criteria are also explained in Section 4.4.

5.5 Results

We present the results visually in terms of scatter plots and bargraphs. A complete set of tables, including p-values, is available in Appendix B.1. Also in Appendix B.1 it is explained how the p-values are obtained.

Figures 5.1 and 5.2 (scatter and histograms plots respectively) compare the performance of BNs and GGMs on the synthetic Gaussian data and the protein concentrations from the cytometry experiment. The two panels on the top of both figures refer to the Gaussian data. Without interventions, BNs and GGMs achieve a similar performance in terms of both AUC and TP scores. Interventions lead to

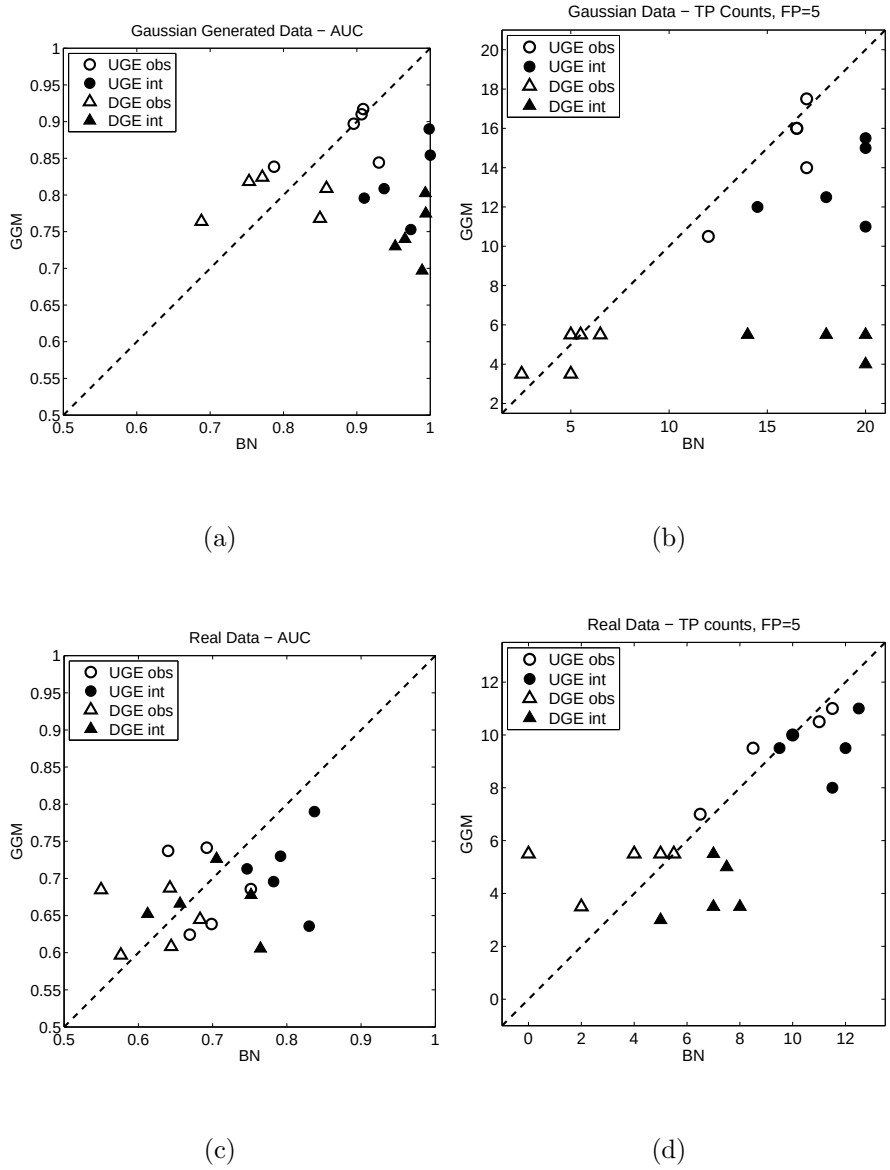


Figure 5.1: **GGMs versus BNs on Gaussian and cytometry data.** Scatterplots comparing the performance of GGMs (vertical axis) with BNs (horizontal axis). The diagonal line represents equal performance. Symbols above that line indicate that GGMs outperform BNs. Conversely, symbols below that line point to a better performance of BNs over GGMs. Each subfigure compares the results obtained from two different data types, using only passive observations (empty symbols) and including active interventions (filled symbols). Two different evaluation criteria have been applied, based on directed graphs (DGE, represented by triangles) and their undirected skeletons (UGE, represented by circles). The four panels refer to different data and scoring criteria. *Top:* Gaussian data, AUC score (left) and TP counts (right). *Bottom:* Cytometry data, AUC score (left) and TP counts (right).

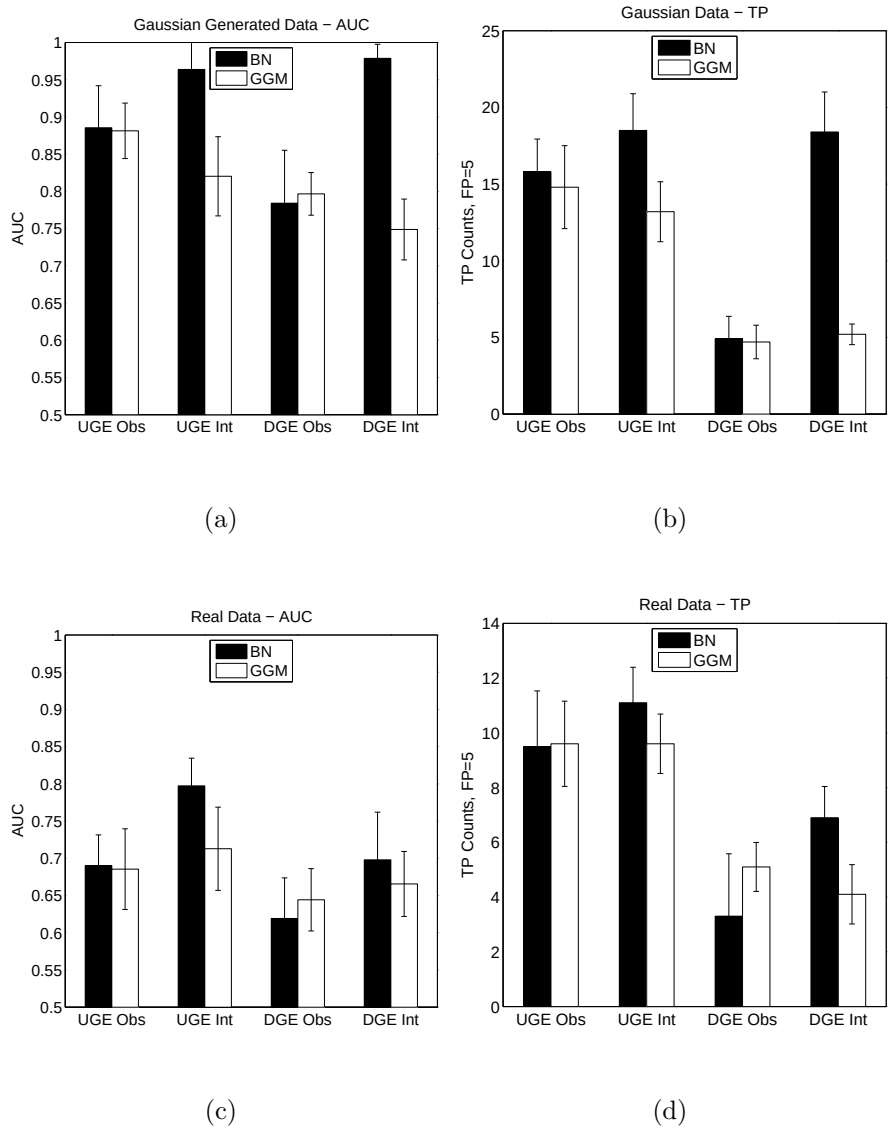


Figure 5.2: **GGMs versus BNs on Gaussian and cytometry data histograms.** Histograms showing the average AUC scores and TP counts for BNs (filled bars) and GGMs (empty bars). The error bars are the standard deviation measured over the 5 different data sets. The codes under the histograms indicate the type of evaluation (UGE versus DGE) and whether observational (Obs) or interventional (Int) data have been used. *Top:* Gaussian data, AUC score (left) and TP counts (right). *Bottom:* Cytometry data, AUC score (left) and TP counts (right).

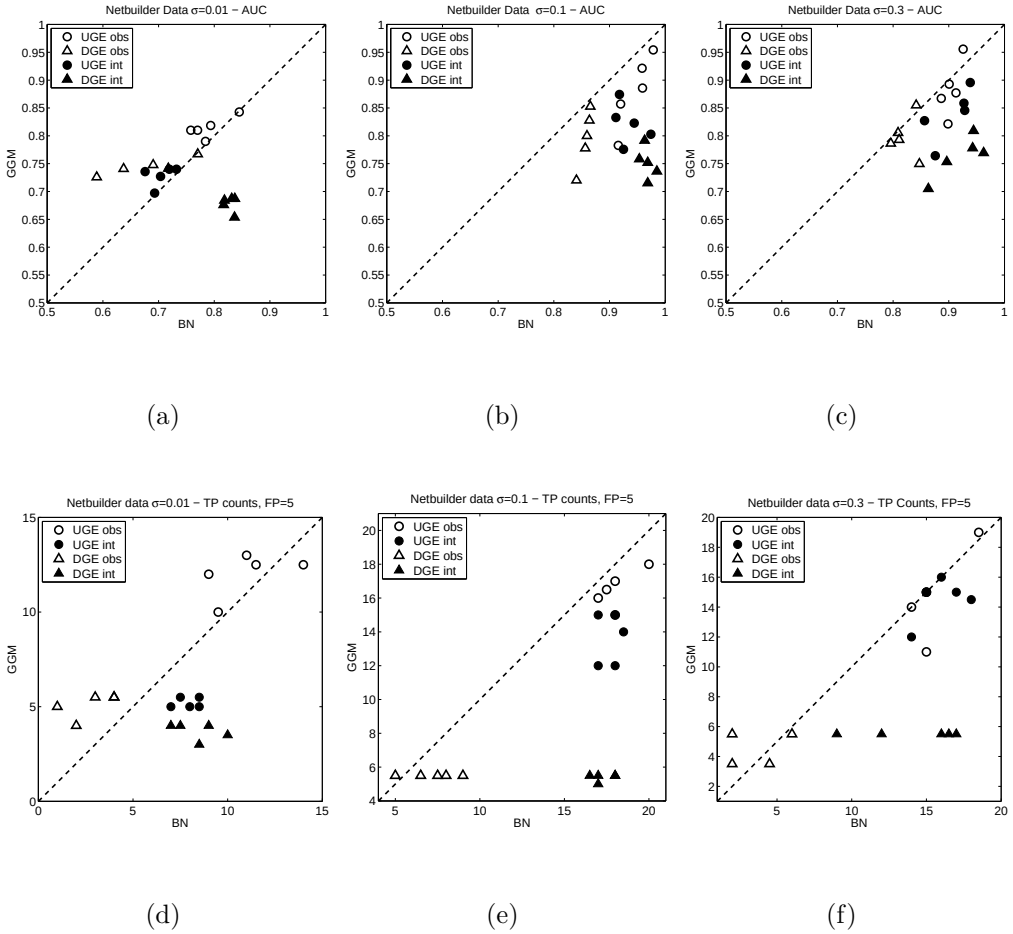


Figure 5.3: **GGMs versus BNs on data simulated with Netbuilder.** This figure compares the performance of GGMs and BNs on the synthetic data generated with Netbuilder. The columns refer to different standard deviations of the additive Gaussian noise. *Left column:* $\sigma = 0.01$. *Centre column:* $\sigma = 0.1$. *Right column:* $\sigma = 0.3$. The two rows refer to different scoring criteria, discussed in Section 5.4. *Top row:* AUC score. *Bottom row:* TP count. The subfigures in the six panels show scatterplots of GGM scores plotted against BN scores; a detailed explanation of the symbols is given in the caption of Figure 5.1.

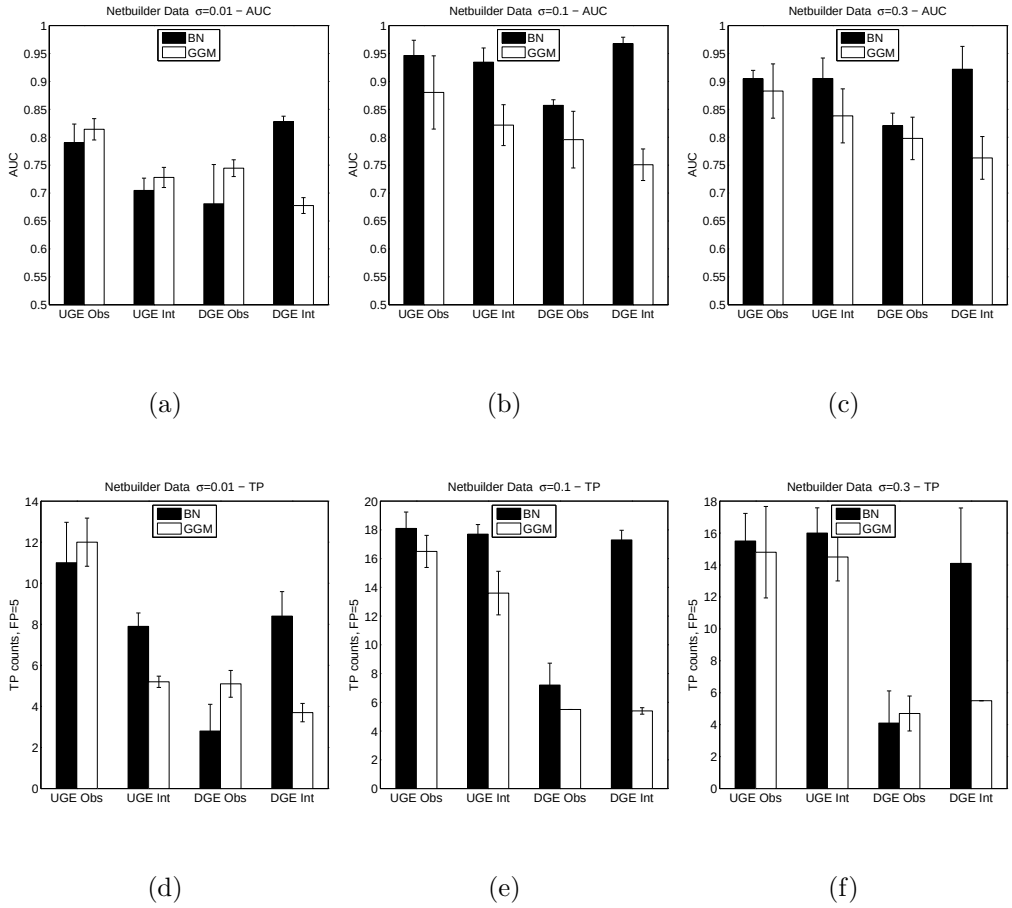


Figure 5.4: **GGMs versus BNs on data simulated with Netbuilder.** This figure compares the performance of GGMs and BNs on the synthetic data generated with Netbuilder. The columns refer to different standard deviations of the additive Gaussian noise. *Left column:* $\sigma = 0.01$. *Centre column:* $\sigma = 0.1$. *Right column:* $\sigma = 0.3$. The two rows show histograms with average AUC scores and TP counts for BNs (filled bars) and GGMs (empty bars); see the caption of Figure 5.2 for further explanations. In each panel, the two rows refer to different scoring criteria, discussed in Section 5.4. *Top row:* AUC score. *Bottom row:* TP count.

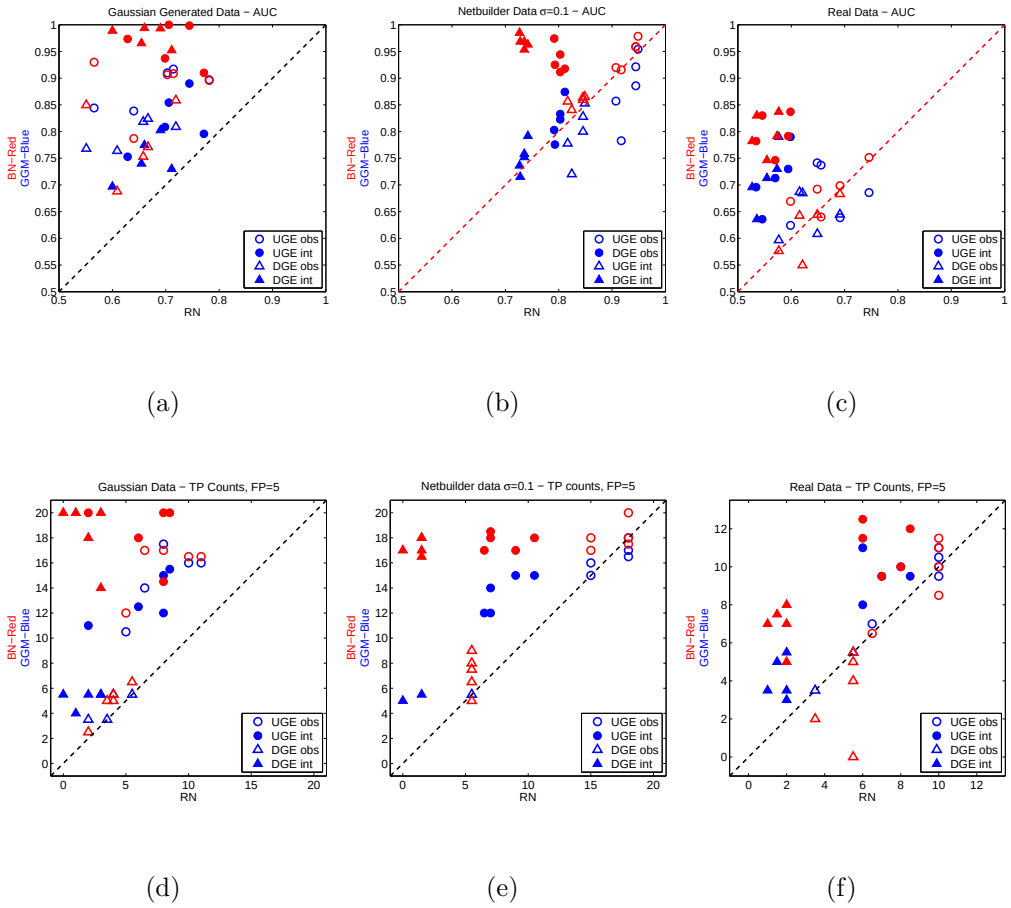


Figure 5.5: **GGMs and BNs versus RNs.** This figure compares the performance of GGMs and BNs (vertical axis) with RNs (horizontal axis). The columns refer to different data sets. *Left column:* Gaussian data. *Centre column:* Data generated with Netbuilder, subject to additive Gaussian noise with $\sigma = 0.1$. *Right column:* Cytometry data. The two rows refer to different scoring criteria, discussed in Section 5.4. *Top row:* AUC score. *Bottom row:* TP count. The symbols of the six scatterplots are explained in the caption of Figure 5.1. The colours refer to different comparisons. *Red:* BNs versus RNs. *Blue:* GGMs versus RNs.

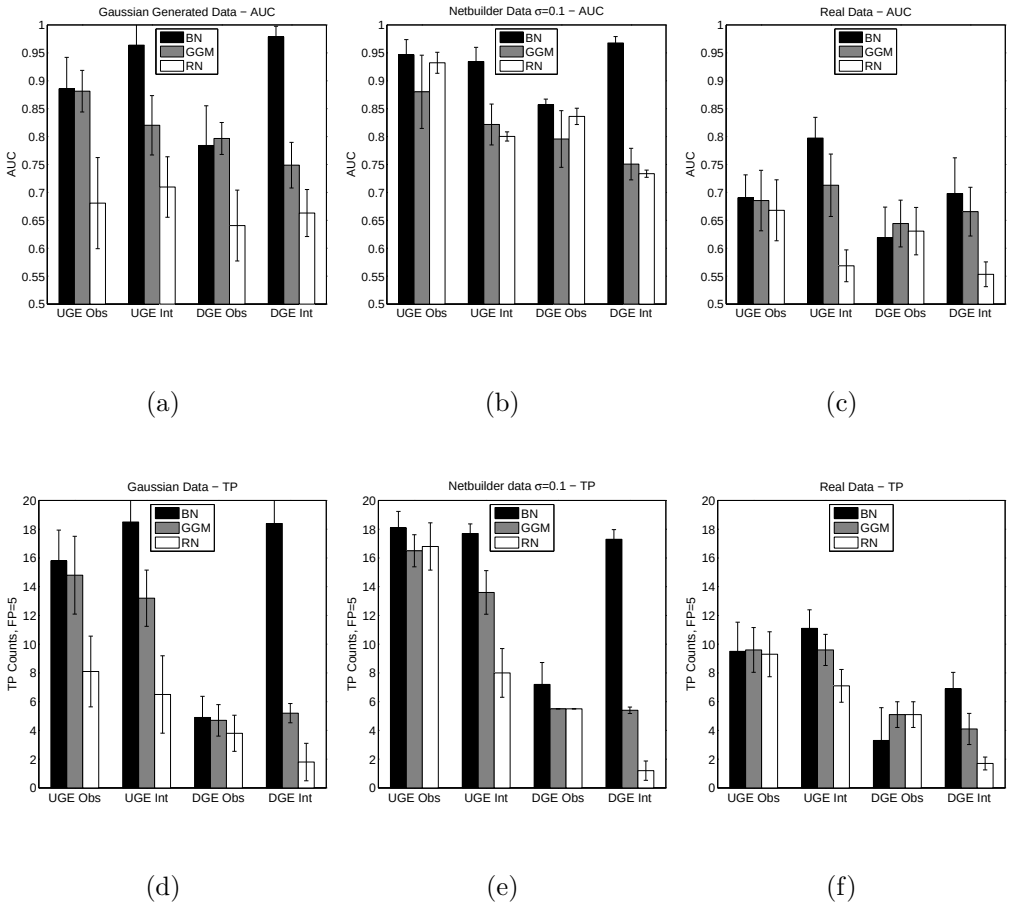


Figure 5.6: **Cross-data comparison between BNs, GGMs and RNs.** The histograms show the average AUC scores and TP counts for BNs (black bars), GGMs (grey bars) and RNs (white bars). The codes under the histograms indicate the type of evaluation (UGE versus DGE) and whether observational (Obs) or interventional (Int) data have been used. The columns refer to different data sets. *Left column:* Gaussian data. *Centre column:* Data generated with Netbuilder, subject to additive Gaussian noise with $\sigma = 0.1$. *Right column:* Cytometry data. The two rows refer to different scoring criteria, discussed in Section 5.4. *Top row:* AUC score. *Bottom row:* TP count.

improved predictions with BNs. As a consequence of interventions, the number of correctly predicted undirected edges increases slightly from 15.8 to 18.5; this is not significant, though ($p = 0.097$). However, the number of correctly predicted directed edges shows a significant increase from 4.9 to 18.4 ($p < 10^{-4}$). On the intervened data, BNs outperform GGMs, and this improvement is significant when the edge directions are taken into account (AUC: $p = 0.0002$, TP: $p = 0.0005$).

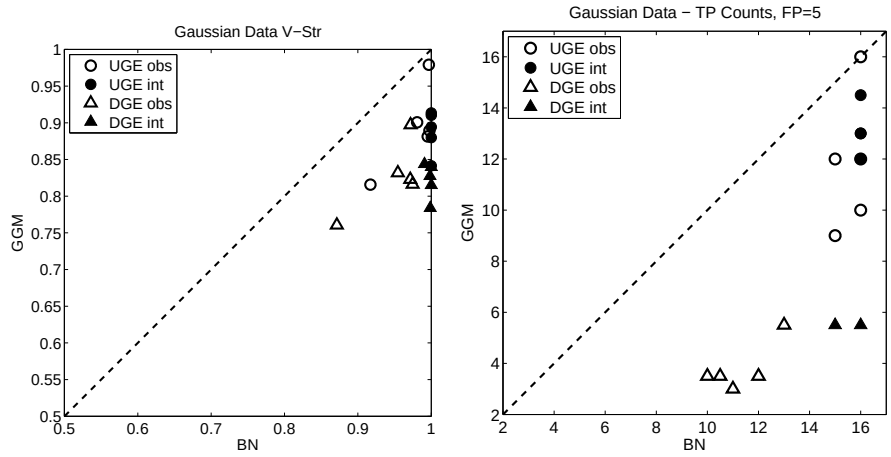
The two panels on the bottom of Figures 5.1 and 5.2 summarize the results obtained for the cytometry data. Without interventions, GGMs and BNs show a similar performance. As a consequence of interventions, the performance of BNs improves, but less substantially than for the Gaussian data. For instance, the number of correctly predicted directed edges increases from 3.3 to 6.9, which is significant ($p = 0.013$). With interventions, BNs tend to outperform GGMs. This improvement is only significant for the DGE-TP score, though ($p = 0.007$); while the UGE-AUC score for BNs is consistently better than for GGMs, its p-value of 0.055 is above the standard significance threshold.

To obtain a deeper understanding of the models' performance, we applied them to the nonlinear simulated data (Netbuilder) with different noise levels. The results are shown in Figures 5.3 and 5.4 (scatter and histograms plots respectively). When comparing the performance of BNs and GGMs on observational data, we observe the following trend. For low noise levels, GGMs slightly outperform BNs, although this difference is only significant for the DGE-TP score ($p = 0.008$); all other p-values are above 0.05. When increasing the noise level, the situation is reversed. BNs outperform GGMs, and the differences are significant for all scores except for DGE-TP (UGE-AUC: $p = 0.025$, DGE-AUC: $p = 0.029$, UGE-TP: $p = 0.016$, DGE-TP: $p = 0.067$). For large noise levels, GGMs and BNs show a similar performance, without a significant difference in any score. Interventions lead to an improvement in the performance of BNs when taking

the edge direction into account. The improvement is significant in both scores, DGE-TP and DGE-AUC, for all noise levels, with $p < 0.002$. The improvement is most pronounced for the medium noise level, where the number of correctly predicted edges increases from 7.2 to 17.3 ($p \ll 10^{-4}$). A comparison between GGMs and BNs reveals that with interventions, BNs consistently outperform GGMs when taking the edge direction into account; all differences are significant with $p < 0.005$.

Figures 5.5 and 5.6 (scatter and histograms plots respectively) compare the performance of BNs and GGMs with RNs. On the Gaussian observational data, both GGMs and BNs consistently outperform RNs. However, there is no significant difference in the performance of the methods on the nonlinear simulated data (Netbuilder) and the cytoflow protein concentrations when no interventions are used; in fact, the DGE-TP scores for BNs are actually worse than those obtained with RNs (see next section for a discussion). With interventions, GGMs outperform RNs on the cytometry data (UGE: $p = 0.001$, DGE: $p = 0.001$), and they obtain higher TP counts than RNs on the nonlinear simulated data ($p < 0.0002$ for both UGE and DGE). BNs consistently outperform RNs on all data sets with respect to all scoring schemes when interventions are used ($p < 0.001$).

In Figures 5.7, 5.8 and 5.9 we show the results that we obtain when executing all the simulations using the synthetic Gaussian and Netbuilder data generated from the v-structure network (Section 4.2.1). Since undirected graphs intrinsically cannot represent v-structures, as discussed in Section 5.1, we would expect an increase in the performance of BNs relative to GGMs. The findings were, overall, similar to the results obtained on the original network. However, the comparison of BNs versus GGMs on observational data showed, in fact, a slight yet significant shift in favour of BNs ($p < 0.05$). This suggests that for networks rich in v-structures, BNs have a systematic advantage over GGMs, in confirmation of our hypothesis. On the observational linear-Gaussian data, the comparison of BNs



(a) (b)

Figure 5.7: **GGMs vs. BNs on Gaussian V-structure data.** Scatter plots comparing the performance of GGMs (vertical axis) with BNs (horizontal axis). The diagonal line represents equal performance. Symbols above the line indicate that GGMs outperform BNs. Conversely symbols below that line point to a better performance of BNs over GGMs. Each subfigure compares the results obtained from two different data types, using only passive observations (empty symbols) and including active interventions (filled symbols). Two different evaluation criteria have been applied, based on directed graphs (DGE, represented by triangles) and their undirected skeletons (UGE, represented by circles). The two panels refer to two different scoring criteria. Left: AUC scores. Right: TP counts.

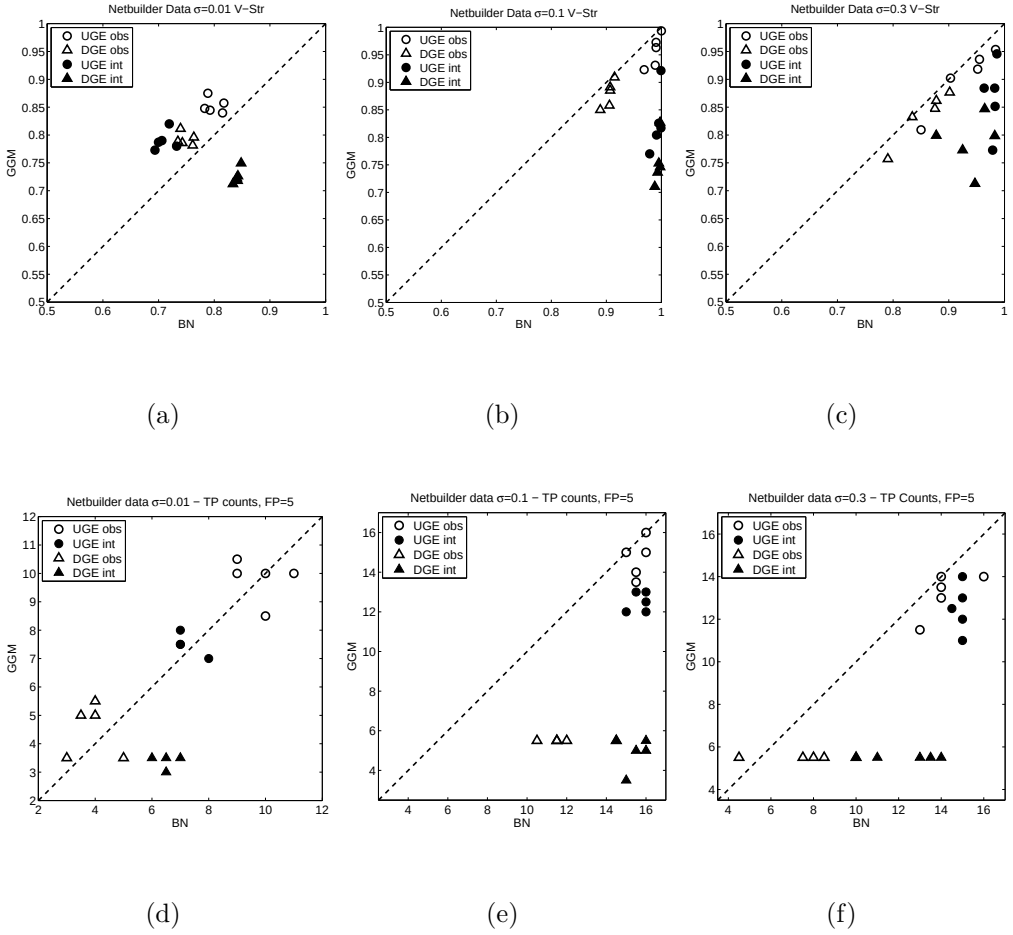


Figure 5.8: **GGMs vs. BNs on Netbuilder V-structure data.** This figure compares the performance of GGMs and BNs on the synthetic data generated with Netbuilder, for the topology with some edges removed. The columns refer to different standard deviations of the additive Gaussian noise. *Left column* $\sigma = 0.01$, *Centre column* $\sigma = 0.1$, *Right column* $\sigma = 0.3$. The two rows refer to different scoring criteria. *Top row:* AUC score. *Bottom row:* TP counts. A detailed explanation of the symbols is given in the caption of figure 5.7

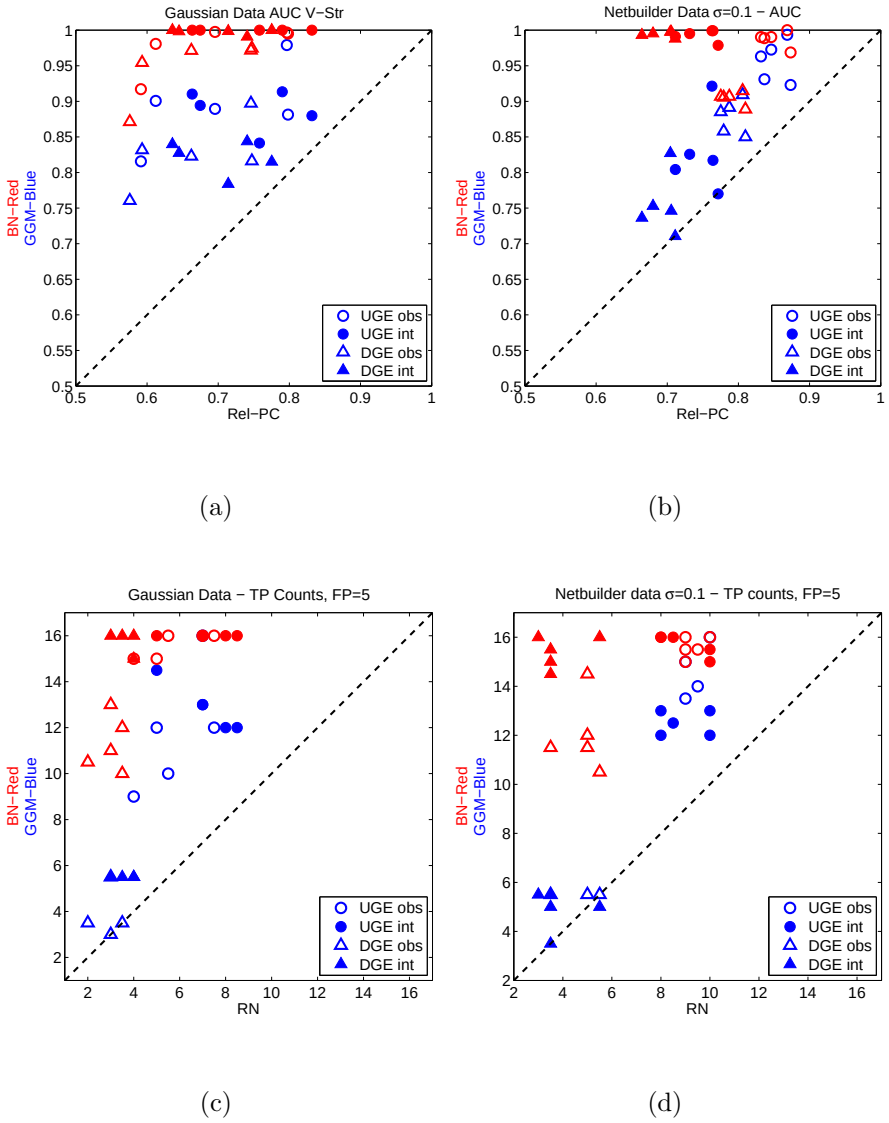


Figure 5.9: **GGMs and BNs vs. RNs. V-structure data.** This figure compares the performance of GGMs and BNs (vertical axis) with RNs (horizontal axis). The columns refer to different data sets. *Left column:* Gaussian data. *Right column:* Netbuilder data with additive Gaussian noise with $\sigma = 0.1$. The two rows refer to different scoring criteria. *Top row:* AUC score. *Bottom row:* TP counts. A detailed explanation of the symbols is given in the caption of figure 5.7. The colours refer to different comparisons. *Red:* BNs versus RNs. *Blue:* GGMs versus RNs

	Gaussian			
	Obs		Int	
	DGE	UGE	DGE	UGE
BN	0.74	0.82	0.93	0.89
GGM	0.77	0.89	0.76	0.87
RN	0.49	0.49	0.53	0.54

(a)

	Netbuilder			
	Obs		Int	
	DGE	UGE	DGE	UGE
BN	0.78	0.86	0.88	0.78
GGM	0.74	0.84	0.72	0.82
RN	0.54	0.55	0.55	0.57

(b)

	Real			
	Obs		Int	
	DGE	UGE	DGE	UGE
BN	0.70	0.79	0.77	0.74
GGM	0.79	0.93	0.72	0.81
RN	0.77	0.88	0.57	0.60

(c)

Figure 5.10: **Separation Scores.** The separation score is defined as $S = T/(T + F)$, where T is the average score of a true edge, and F is the average score of a false edge. The perfect separation score of $S = 1$ is obtained when assigning a zero score to all false edges. Conversely, a method that cannot distinguish between true and false edges leads to an average separation score of $S = 0.5$. The abbreviations *Obs* and *Int* refer to observational and interventional data, respectively. The numbers are averages over all simulations carried out in the indicated category.

versus GGMs showed a significant shift in favour of BNs, with $p < 0.05$ for all performance scores; this confirms our hypothesis. There was no significant difference between the performance scores of BNs and GGMs on the nonlinear data generated with Netbuilder, though.

5.6 Discussion

5.6.1 Dependence on the noise level

When varying the noise level on the nonlinear simulated data (Figures 5.3 and 5.4) we observe that when increasing the noise level, the performance with BNs first increases, and then decreases. For instance, the average number of predicted true undirected edges increases from $TP = 11$ for $\sigma = 0.01$ to $TP = 18$ for $\sigma = 0.1$, and then decreases again to $TP = 15.5$ for $\sigma = 0.3$. To understand this behaviour, consider a parent node that regulates several children, where the children do not have any direct interactions; see Figure 3.12(b). Without noise, the response of each child is a deterministic function of the parent. However, this implies a deterministic functional relationship between the children. Consequently, the true network cannot be distinguished from a network in which all children are connected by edges, and it is intrinsically impossible to learn the true network. The deterministic relationship between the children is destroyed by the addition of noise, which renders, on average, the signal of a child more similar to that of its parent than that of a sibling. Consequently, some noise is useful and forms the basis for learning gene regulatory networks from data. However, when the noise level becomes so large that it hides the regular signal, successful learning will no longer be feasible. Hence, we would expect the accuracy of reconstructing regulatory networks to first increase and then decrease with increasing noise level, and this trend is confirmed in our simulations.

5.6.2 GGMs versus RNs

To better understand the different performance of GGMs and RNs, we computed the average association scores for true edges and non-edges. The separation score is defined in the caption of Table 5.10. The results are shown in Table 5.10 and suggest that GGMs show a clearer separation of the true and false edges than RNs.

This difference has not translated itself into an improved performance of GGMs over RNs in terms of AUC and TP scores for the unintervened non-Gaussian data. The reason is that although the separation between the scores is poorer for RNs than for GGMs, it has not affected the ranking of the edges. However, this finding suggests that inference with RNs is less stable than with GGMs. In fact, for interventions, RNs show a more substantial degradation in their performance than GGMs; GGMs consistently outperform RNs on the intervened cytoflow data ($p < 0.021$), and obtain significantly higher TP counts on the nonlinear simulated data ($p < 10^{-4}$).

5.6.3 Interventions for low noise level.

Figures 5.3(d) and 5.4(d) reveal a curious finding: on interventions, the UGE score for BNs deteriorates. As discussed above, the ability to suppress spurious associations between unconnected nodes deteriorates for low noise levels. Interventions reduce the average noise level; so if the noise is already very low, this further reduction in the noise may lead to the prediction of spurious associations. The deterioration of the UGE (as opposed to the DGE) score can be explained by the fact that a spurious undirected edge is equivalent to two spurious directed edges (since there are twice as many directed as undirected edges in the graph), and that the UGE score does not benefit from any corrections of edge directions that result from the interventions.

5.6.4 Learning directed graphs from the cytometry data

Our analysis reveals an interesting observation for the cytoflow data (Figures 5.1(c,d) and 5.2(c,d)). While interventions lead to an improvement in the performance of BNs, this improvement is more pronounced for the undirected skeleton (UGE score) than the directed graph (DGE score). For instance, in Figure 5.2(c) we observe that BNs outperform GGMs on interventional data

in terms of the UGE AUC scores, but not the DGE AUC scores. Interestingly, a recent study (Dougherty et al., 2005) carried out after the publication of Sachs et al. (2005) reports evidence for a negative feedback loop from Erk1/2 back to Raf, which is not included in the assumed gold standard network taken from Sachs et al. (2005). A negative feedback is known to lead to a stabilization of the output, which compensates for the effect of an intervention on the output path (here: Mek1/2). Hence, this intervention may no longer allow us to resolve the ambiguity about the edge directions. Moreover, the existence of a hidden feedback loop acting on a putative feedforward path may lead to some systematic error in the edge directions, as all methods investigated in the present paper are intrinsically restricted to the modelling of systems without recurrent loops. This example points to a fundamental problem inherent to any evaluation based solely on real biological data, namely, that the underlying true regulatory network is ultimately unknown, and that published "gold-standard" networks have to be taken with caution. For this reason we believe that the analysis carried out for the present comparison, which combines data from a real laboratory experiments with various synthetic and simulated data, will lead to a deeper insight and better understanding than what could be obtained from real laboratory data alone.

5.7 Conclusion

Our main findings can be summarized as follows. BNs and GGMs tend to outperform RNs, but the difference is less pronounced for the nonlinear simulated data (Netbuilder) and the measured protein concentrations (cytometry experiments) than for Gaussian data. Also, there is insufficient evidence for any significant difference between BNs and GGMs on observational data. These findings are different from those reported in Pournara (2005), which seems to result from the improved inference algorithm for GGMs (Schäfer and Strimmer, 2005b). However,

for interventional data, BNs clearly outperform GGMs and RNs when taking the edge directions (DGE score) rather than just the skeletons of the graphs (UGE score) into account. This suggests that the higher computational costs of inference with BNs over GGMs and RNs are not justified for passive observations, but that active interventions in the form of gene knockouts and over-expressions are required to exploit the full potential of BNs. As another possibility for exploring the full potential of BNs one can consider the use of extra sources of information as prior biological knowledge as discussed in the next chapter.

Chapter 6

Combining prior biological knowledge with gene expression

This chapter presents results of a published journal paper (Werhli and Husmeier, 2007) and a conference paper (Husmeier and Werhli, 2007).

6.1 Introduction

An important and challenging problem in systems biology is the inference of gene regulatory networks from high-throughput microarray expression data. Various machine learning and statistical methods have been applied to this end, like Bayesian Networks (BNs) (Friedman et al., 2000), Relevance Networks (Butte and Kohane, 2003) and Graphical Gaussian Models (Schäfer and Strimmer, 2005b). An intrinsic difficulty with these approaches is that complex interactions involving many genes have to be inferred from sparse and noisy data. This leads to a poor reconstruction accuracy and suggests that the inclusion of complementary information is indispensable (Husmeier, 2003). A promising approach in this direction has been proposed by Imoto et al. (2003a). The authors formulate the learning scheme in a Bayesian framework. This scheme allows the systematic integration of gene expression data with biological knowl-

edge from other types of postgenomic data or the literature via a prior distribution over network structures. The hyperparameters of this distribution are inferred together with the network structure in a maximum a posteriori sense by maximizing the joint posterior distribution with a heuristic greedy optimization algorithm. As prior knowledge, the authors extracted protein-DNA interactions from the Yeast Proteome Database. The framework has subsequently been applied to a variety of different sources of biological prior knowledge, where gene regulatory networks were inferred from a combination of gene expression data with transcription factor binding motifs in promoter sequences (Tamada et al., 2003), protein-protein interactions (Nariai et al., 2004), evolutionary information (Tamada et al., 2005), and pathways from the KEGG database (Imoto et al., 2006). In this chapter this work is complemented in various respects.

First, we adopt a sampling-based approach to Bayesian inference as opposed to the optimization schemes applied in the work cited above. The latter aims to find the network structure and the hyperparameters that maximize the joint posterior distribution. This approach is appropriate for posterior distributions that are sharply peaked. However, when gene expression data are sparse and noisy and the prior knowledge is susceptible to intrinsic uncertainty as well, this condition is unlikely to be met. In that case, it is more appropriate to follow Madigan and York (1995), Giudici and Castelo (2003) and Friedman and Koller (2003) and sample network structures from the posterior distribution with Markov chain Monte Carlo (MCMC). We pursue the same approach, and additionally sample the hyperparameters associated with the prior distribution from the joint posterior distribution with MCMC.

Second, we aim to obtain a deeper understanding of the proposed modelling and inference scheme. The prior distribution proposed in Imoto et al. (2003a) takes the form of a Gibbs distribution, in which the prior knowledge is encoded via an energy function, and an inverse temperature hyperparameter determines

the weight that is assigned to it. In our study, we have designed a scenario in which the energy takes on a particular form such that computing the marginal posterior distribution over the hyperparameter becomes analytically tractable. This closed-form expression is compared with MCMC simulations on simulated and real-world data for the more general scenario in which the marginal posterior distribution is intractable, elucidating various aspects of the modelling approach.

Third, we extend the approach of Imoto et al. (2003a) to include more than one energy function. This approach allows the simultaneous inclusion of different sources of prior knowledge, like promoter motifs and KEGG pathways, each modelled by a separate energy. Each energy function is associated with its own hyperparameter. All hyperparameters are sampled from the posterior distribution with MCMC. In this way, the relative weights related to the different sources of prior knowledge are consistently inferred within the Bayesian context, automatically trading off their relative influences in light of the data.

Fourth, we provide a set of independent evaluations of the viability of the Bayesian inference scheme on various synthetic and real-world data, thereby complementing the results of the studies referred to above. In particular, we apply the proposed method to the integration of two independent sources of transcription factor binding locations from immunoprecipitation experiments with microarray gene expression data from the yeast cell cycle, and the integration of KEGG pathways with cytometry experiments for determining protein interactions related to the Raf signalling pathway.

This chapter is organized as follows. In Section 6.2 we refer to the methodology of Bayesian networks and present the proposed Bayesian approach to integrating biological prior knowledge into the inference scheme. In Section 6.3 we investigate the behaviour of the proposed inference scheme on an idealized population of network structures, for which a closed-form expression of the relevant posterior distribution can be obtained. Section 6.4 presents the synthetic and real data

sets that we used for evaluating the performance of the proposed method. The results from this method are presented in Section 6.5. In Section 6.6 we introduce a modified version of the method, apply it to the data sets discussed earlier and present the results obtained with this modified version of the algorithm. Finally, a concluding discussion is presented in Section 6.7.

6.2 Methodology

The methodology of static Bayesian networks and dynamic Bayesian networks are presented in Section 3.2 and in Section 3.3 respectively.

6.2.1 Biological prior knowledge

As mentioned in the Introduction section, the objective of the present work is to study the integration of biological prior knowledge into the inference of gene regulatory networks. To this end, we need to define a function that measures the agreement between a given network $\mathcal{M}$ and the biological prior knowledge that we have at our disposal. We follow the approach proposed by Imoto et al. (2003a) and call this measure the energy E, borrowing the name from the statistical physics community.

6.2.1.1 The energy of a network

A network $\mathcal{M}$ is represented by a binary adjacency matrix, where each entry $\mathcal{M}_{ij}$ can be either 0 or 1. A zero entry, $\mathcal{M}_{ij} = 0$, indicates the absence of an edge between node_i and node_j. Conversely if $\mathcal{M}_{ij} = 1$ there is a directed edge from node_i to node_j. We define the biological prior knowledge matrix B to be a matrix in which the entries $B_{ij} \in [0, 1]$ represent our knowledge about interactions between nodes as follows:

- If entry $B_{ij} = 0.5$, we do not have any prior knowledge about the presence or absence of the directed edge between node$_i$ and node$_j$.
- If $0 \leq B_{ij} < 0.5$ we have prior evidence that there is no directed edge between node$_i$ and node$_j$. The evidence is stronger as B_{ij} is closer to 0.
- If $0.5 < B_{ij} \leq 1$ we have prior evidence that there is a directed edge pointing from node$_i$ to node$_j$. The evidence is stronger as B_{ij} is closer to 1.

Note that despite their restriction to the unit interval, the B_{ij} are not probabilities in a stochastic sense. To obtain a proper probability distribution over networks, we have to introduce an explicit normalization procedure, as will be discussed shortly.

Having defined how to represent a network $\mathcal{M}$ and the biological prior knowledge B, we can now define the 'energy' of a network:

$$E(\mathcal{M}) = \sum_{i,j=1}^{N} |B_{i,j} - \mathcal{M}_{i,j}| \tag{6.1}$$

where N is the total number of nodes in the studied domain. The energy E is zero for a perfect match between the prior knowledge B and the actual network structure $\mathcal{M}$, while increasing values of E indicate an increasing mismatch between B and $\mathcal{M}$.

6.2.1.2 One source of biological prior knowledge

To integrate the prior knowledge expressed by Equation 6.1 into the inference procedure, we follow Imoto et al. (2003a) and define the prior distribution over network structures $\mathcal{M}$ to take the form of a Gibbs distribution:

$$P(\mathcal{M}|\beta) = \frac{e^{-\beta E(\mathcal{M})}}{Z(\beta)} \tag{6.2}$$

where the energy $E(\mathcal{M})$ was defined in Equation 6.1, β is a hyperparameter that corresponds to an inverse temperature in statistical physics, and the denominator

is a normalizing constant that is usually referred to as the partition function:

$$Z(\beta) = \sum_{\mathcal{M} \in \mathbb{M}} e^{-\beta E(\mathcal{M})} \tag{6.3}$$

Note that the summation extends over the set of all possible network structures $\mathcal{M}$. The hyperparameter β can be interpreted as a factor that indicates the strength of the influence of the biological prior knowledge relative to the data. For $\beta \to 0$, the prior distribution defined in Equation 6.2 becomes flat and uninformative about the network structure. Conversely, for $\beta \to \infty$, the prior distribution becomes sharply peaked at the network structure with the lowest energy.

For DBNs we can exploit the modularity of Bayesian networks and compute the sum in Equation 6.3 efficiently. Note that $E(\mathcal{M})$ in Equation 6.1 can be rewritten as follows:

$$E(\mathcal{M}) = \sum_{n=1}^{N} \mathcal{E}\left(n, \pi_{\mathcal{M}}(n)\right) \tag{6.4}$$

where $\pi_{\mathcal{M}}(n)$ is the set of parents of node n in the graph $\mathcal{M}$, and we have defined:

$$\mathcal{E}\left(n, \pi_{\mathcal{M}}(n)\right) = \sum_{i \in \pi_{\mathcal{M}}(n)} (1 - B_{in}) + \sum_{i \notin \pi_{\mathcal{M}}(n)} B_{in} \tag{6.5}$$

Inserting Equation (6.4) into Equation (6.3) we obtain:

$$\begin{aligned} Z &= \sum_{\mathcal{M} \in \mathbb{M}} e^{-\beta E(\mathcal{M})} \\ &= \sum_{\pi_{\mathcal{M}}(1)} \cdots \sum_{\pi_{\mathcal{M}}(N)} e^{-\beta(\mathcal{E}(1,\pi_{\mathcal{M}}(1)) + \ldots + \mathcal{E}(N,\pi_{\mathcal{M}}(N)))} \\ &= \prod_{n} \sum_{\pi_{\mathcal{M}}(n)} e^{-\beta \mathcal{E}(n,\pi_{\mathcal{M}}(n))} \end{aligned} \tag{6.6}$$

Here, the summation in the last equation extends over all parent configurations $\pi_{\mathcal{M}}(n)$ of node n, which in the case of a fan-in restriction is subject to constraints on their cardinality. Note that the essence of Equation (6.6) is a dramatic reduction in the computational complexity. Rather than summing over the whole space of network structures, whose cardinality increases super-exponentially with

the number of nodes N, we only need to sum over all parent configurations of each node; the complexity of this operation is $\binom{N-1}{m}$(where m is the maximum fan-in), that is, polynomial in N. The reason for this simplification is the fact that any modification of the parent configuration of a node in a DBN leads to a new valid DBN by construction. This convenient feature does not apply to static BNs, though, where modifications of a parent configuration $\pi_{\mathcal{M}}(n)$ may lead to directed cyclic structures, which are invalid and hence have to be excluded from the summation in Equation (6.6). The detection of directed cycles is a global operation. This destroys the modularity inherent in Equation (6.6), and leads to a considerable explosion of the computational complexity. Note, however, that Equation (6.6) still provides an upper bound on the true partition function. When densely connected graphs are ruled out by a fan-in restriction, as commonly done, the number of cyclic terms that need to be excluded from Equation (6.6) can be assumed to be relatively small. We can then expect the bound to be rather tight, as suggested by Imoto et al. (2006), and use it to approximate the true partition function. In all our simulations we assumed a fan-in restriction of three, as has widely been applied by different authors; e.g. Friedman et al. (2000); Friedman and Koller (2003); Husmeier (2003). We tested the viability of the approximation made for static Bayesian networks in our simulations, to be discussed in Section 6.5; see especially Figures 6.14 and 6.15.

6.2.1.3 Multiple sources of biological prior knowledge

The method described in the previous section can be generalized to multiple sources of prior knowledge. To keep the notation transparent, we restrict our discussion to two sources of prior knowledge; an extension to more than two sources is straightforward and follows along the same line of argumentation as presented here. We assume that the biological prior knowledge from each independent source is represented by a separate prior knowledge matrix B^k, $k \in \{1, 2\}$, each

satisfying the requirements laid out in the previous section. This gives us two energy functions:

$$E_1(\mathcal{M}) = \sum_{i,j=1}^{N} \left| B^1_{i,j} - \mathcal{M}_{i,j} \right| \tag{6.7}$$

$$E_2(\mathcal{M}) = \sum_{i,j=1}^{N} \left| B^2_{i,j} - \mathcal{M}_{i,j} \right| \tag{6.8}$$

where each energy is associated with its own hyperparameter β_k. The prior probability of a network $\mathcal{M}$ given the hyperparameters β_1 and β_2 is now defined as:

$$P(\mathcal{M}|\beta_1, \beta_2) = \frac{e^{-\{\beta_1 E_1(\mathcal{M}) + \beta_2 E_2(\mathcal{M})\}}}{Z(\beta_1, \beta_2)} \tag{6.9}$$

where the partition function in the denominator is given by:

$$Z(\beta_1, \beta_2) = \sum_{\mathcal{M} \in \mathbb{M}} e^{-\{\beta_1 E_1(\mathcal{M}) + \beta_2 E_2(\mathcal{M})\}} \tag{6.10}$$

For DBNs, the partition function can again be efficiently computed in closed form. Similarly to the discussion above Equation (6.6), we can rewrite Equations (6.7) and (6.8) as follows:

$$E_1(\mathcal{M}) = \sum_{n=1}^{N} (n, \pi_{\mathcal{M}}(n)) \tag{6.11}$$

$$E_2(\mathcal{M}) = \sum_{n=1}^{N} (n, \pi_{\mathcal{M}}(n)) \tag{6.12}$$

where $\pi_{\mathcal{M}}(n)$ is the set of parents of node n in the graph $\mathcal{M}$, and we have defined:

$$\mathcal{E}_1 (n, \pi_{\mathcal{M}}(n)) = \sum_{i \in \pi_{\mathcal{M}}(n)} \left(1 - B^1_{in}\right) + \sum_{i \notin \pi_{\mathcal{M}}(n)} B^1_{in} \tag{6.13}$$

$$\mathcal{E}_2 (n, \pi_{\mathcal{M}}(n)) = \sum_{i \in \pi_{\mathcal{M}}(n)} \left(1 - B^2_{in}\right) + \sum_{i \notin \pi_{\mathcal{M}}(n)} B^2_{in} \tag{6.14}$$

$$\tag{6.15}$$

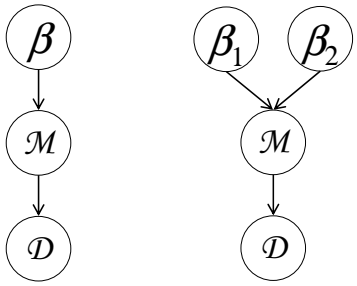


Figure 6.1: **Probabilistic graphical models.** The two probabilistic graphical models represent conditional independence relations between the data $\mathcal{D}$, the network structure $\mathcal{M}$, and the hyperparameters of the prior on $\mathcal{M}$. The left graph shows the situation of a single source of prior knowledge, with one hyperparameter β. The graph in the right panel shows the situation of two independent sources of prior knowledge, associated with two separate hyperparameters β_1 and β_2. The conditional independence relations can be obtained from the graphs according to the standard rules of factorization in Bayesian networks, as discussed, e.g., in Heckerman (1999). This leads to the following expansions. Left panel: $P(\mathcal{D}, \mathcal{M}, \beta) = P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\beta)P(\beta)$. Right panel: $P(\mathcal{D}, \mathcal{M}, \beta_1, \beta_2) = P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\beta_1, \beta_2)P_1(\beta_1)P_2(\beta_2)$.

Inserting Equations (6.11) and (6.12) into Equation (6.10), we obtain:

$$\begin{aligned} Z &= \sum_{\mathcal{M}\in\mathbb{M}} e^{-\{\beta_1 E_1(\mathcal{M})+\beta_2 E_2(\mathcal{M})\}} \\ &= \sum_{\pi_{\mathcal{M}}(1)} \cdots \sum_{\pi_{\mathcal{M}}(N)} e^{-\{\beta_1[\mathcal{E}_1(1,\pi_{\mathcal{M}}(1))+\ldots+\mathcal{E}_1(N,\pi_{\mathcal{M}}(N))]+\beta_2[\mathcal{E}_2(1,\pi_{\mathcal{M}}(1))+\ldots+\mathcal{E}_2(N,\pi_{\mathcal{M}}(N))]\}} \\ &= \prod_n \sum_{\pi_{\mathcal{M}}(n)} e^{-\{\beta_1\mathcal{E}_1(n,\pi_{\mathcal{M}}(n))+\beta_2\mathcal{E}_2(n,\pi_{\mathcal{M}}(n))\}} \end{aligned} \tag{6.16}$$

For static BNs, this expression provides an upper bound, which can be expected to be tight for strict fan-in restrictions; see the discussion below Equation (6.6).

6.2.2 MCMC sampling scheme

Having defined the prior probability distribution over network structures, the next objective is to extend the MCMC scheme of Equation (3.20) on Section 3.2.2.1 to sample both the network structure and the hyperparameters from the posterior distribution.

6.2.2.1 MCMC with one source of biological prior knowledge

Starting from a definition of the prior distribution on the hyperparameter β, $P(\beta)$, our aim is to sample the network structure $\mathcal{M}$ and the hyperparameter β from

the posterior distribution $P(\mathcal{M}, \beta|\mathcal{D})$. To this end, we propose a new network structure $\mathcal{M}_{\text{new}}$ from the proposal distribution $Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})$ and, additionally, a new hyperparameter from the proposal distribution $R(\beta_{\text{new}}|\beta_{\text{old}})$. We then accept this move according to the standard Metropolis-Hastings update rule (Hastings, 1970) with the following acceptance probability:

$$A = \min\left\{\frac{P(\mathcal{D}, \mathcal{M}_{\text{new}}, \beta_{\text{new}})Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})R(\beta_{\text{old}}|\beta_{\text{new}})}{P(\mathcal{D}, \mathcal{M}_{\text{old}}, \beta_{\text{old}})Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})R(\beta_{\text{new}}|\beta_{\text{old}})}, 1\right\} \tag{6.17}$$

which owing to the conditional independence relations depicted in Figure 6.1 can be expanded as follows:

$$A = \min\left\{\frac{P(\mathcal{D}|\mathcal{M}_{\text{new}})P(\mathcal{M}_{\text{new}}|\beta_{\text{new}})P(\beta_{\text{new}})Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})R(\beta_{\text{old}}|\beta_{\text{new}})}{P(\mathcal{D}|\mathcal{M}_{\text{old}})P(\mathcal{M}_{\text{old}}|\beta_{\text{old}})P(\beta_{\text{old}})Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})R(\beta_{\text{new}}|\beta_{\text{old}})}, 1\right\} \tag{6.18}$$

To increase the acceptance probability and, hence, mixing and convergence of the Markov chain, it is advisable to break the move up into two submoves. First, we sample a new network structure $\mathcal{M}_{\text{new}}$ from the proposal distribution $Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})$ while keeping the hyperparameter β fixed, and accept this move with the following acceptance probability:

$$A(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}}) = \min\left\{\frac{P(\mathcal{D}|\mathcal{M}_{\text{new}})P(\mathcal{M}_{\text{new}}|\beta)Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})}{P(\mathcal{D}|\mathcal{M}_{\text{old}})P(\mathcal{M}_{\text{old}}|\beta)Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})}, 1\right\} \tag{6.19}$$

Next, we sample a new hyperparameter β from the proposal distribution $R(\beta_{\text{new}}|\beta_{\text{old}})$ for a fixed network structure $\mathcal{M}$, and accept this move with the following acceptance probability:

$$A(\beta_{\text{new}}|\beta_{\text{old}}) = \min\left\{\frac{P(\mathcal{M}|\beta_{\text{new}})P(\beta_{\text{new}})R(\beta_{\text{old}}|\beta_{\text{new}})}{P(\mathcal{M}|\beta_{\text{old}})P(\beta_{\text{old}})R(\beta_{\text{new}}|\beta_{\text{old}})}, 1\right\} \tag{6.20}$$

For a uniform prior distribution $P(\beta)$ and a symmetric proposal distribution $R(\beta_{\text{new}}|\beta_{\text{old}})$, this expression simplifies:

$$A(\beta_{\text{new}}|\beta_{\text{old}}) = \min\left\{\frac{P(\mathcal{M}|\beta_{\text{new}})}{P(\mathcal{M}|\beta_{\text{old}})}, 1\right\} \tag{6.21}$$

The two submoves are iterated until some convergence criterion is satisfied. See Section 3.2.2.4 on page 42 for a discussion about convergence diagnostics.

The acceptance probability Equation (6.21) can be rewritten as:

$$A(\beta_{\text{new}}|\beta_{\text{old}}) = \min\left\{\frac{e^{-E(\mathcal{M})(\beta_{\text{new}}-\beta_{\text{old}})}Z(\beta_{\text{old}})}{Z(\beta_{\text{new}})}, 1\right\} \tag{6.22}$$

which clearly indicates the dependency of this acceptance probability on the partition functions $Z(\beta_{\text{old}})$ and $Z(\beta_{\text{new}})$. For a discussion about how the value of the partition function is obtained and the approximations that have to be made see the text below Equation 6.6.

6.2.2.2 MCMC with multiple sources of biological prior knowledge

The scheme presented in the previous section can be extended to multiple sources of prior knowledge. To avoid opacity in the notation, we restrict our discussion to two independent sources of prior knowledge. The generalization to more than two sources is straightforward and follows the same principles as discussed in this section. Starting from two prior distributions on the hyperparameters, $P_1(\beta_1)$ and $P_2(\beta_2)$, our objective is to sample network structures and hyperparameters from the posterior distribution $P(\mathcal{M}, \beta_1, \beta_2|\mathcal{D})$. Again, we follow the standard Metropolis-Hastings scheme (Hastings, 1970). We sample a new network structure $\mathcal{M}_{\text{new}}$ from the proposal distribution $Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})$, and new hyperparameters from the proposal distributions $R_1(\beta_{1_{\text{new}}}|\beta_{1_{\text{old}}})$ and $R_2(\beta_{2_{\text{new}}}|\beta_{2_{\text{old}}})$. The acceptance probability of this move is:

$$A = \min\left\{\frac{P(\mathcal{D}, \mathcal{M}_{\text{new}}, \beta_{1_{\text{new}}}, \beta_{2_{\text{new}}})Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})R_1(\beta_{1_{\text{old}}}|\beta_{1_{\text{new}}})R_2(\beta_{2_{\text{old}}}|\beta_{2_{\text{new}}})}{P(\mathcal{D}, \mathcal{M}_{\text{old}}, \beta_{1_{\text{old}}}, \beta_{2_{\text{old}}})Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})R_1(\beta_{1_{\text{new}}}|\beta_{1_{\text{old}}})R_2(\beta_{2_{\text{new}}}|\beta_{2_{\text{old}}})}, 1\right\} \tag{6.23}$$

From the conditional independence relations depicted in Figure 6.1, this expression can be expanded as follows:

$$\begin{aligned} A = \min\Bigg\{&\frac{P(\mathcal{D}|\mathcal{M}_{\text{new}})P(\mathcal{M}_{\text{new}}|\beta_{1_{\text{new}}}, \beta_{2_{\text{new}}})P_1(\beta_{1_{\text{new}}})P_2(\beta_{2_{\text{new}}})}{P(\mathcal{D}|\mathcal{M}_{\text{old}})P(\mathcal{M}_{\text{old}}|\beta_{1_{\text{old}}}, \beta_{2_{\text{old}}})P_1(\beta_{1_{\text{old}}})P_2(\beta_{2_{\text{old}}})} \times \\ &\frac{Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})R_1(\beta_{1_{\text{old}}}|\beta_{1_{\text{new}}})R_2(\beta_{2_{\text{old}}}|\beta_{2_{\text{new}}})}{Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})R_1(\beta_{1_{\text{new}}}|\beta_{1_{\text{old}}})R_2(\beta_{2_{\text{new}}}|\beta_{2_{\text{old}}})}, 1\Bigg\} \end{aligned} \tag{6.24}$$

As discussed in the previous section, it is advisable to break this move up into three submoves:

- Sample a new network structure $\mathcal{M}_{\text{new}}$ from the proposal distribution $Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})$ for fixed hyperparameters β_1 and β_2.
- Sample a new hyperparameter $\beta_{1_{\text{new}}}$ from the proposal distribution $R_1(\beta_{1_{\text{new}}}|\beta_{1_{\text{old}}})$ for fixed hyperparameter β_2 and fixed network structure $\mathcal{M}$.
- Sample a new hyperparameter $\beta_{2_{\text{new}}}$ from the proposal distribution $R_2(\beta_{2_{\text{new}}}|\beta_{2_{\text{old}}})$ for fixed hyperparameter β_1 and fixed network structure $\mathcal{M}$.

Assuming uniform prior distributions $P_1(\beta_1)$ and $P_2(\beta_2)$ as well as symmetric proposal distributions $R_1(\beta_{1_{\text{new}}}|\beta_{1_{\text{old}}})$ and $R_2(\beta_{2_{\text{new}}}|\beta_{2_{\text{old}}})$, the corresponding acceptance probabilities are given by the following expressions:

$$A(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}}) = \min\left\{\frac{P(\mathcal{D}|\mathcal{M}_{\text{new}})P(\mathcal{M}_{\text{new}}|\beta_1,\beta_2)Q(\mathcal{M}_{\text{old}}|\mathcal{M}_{\text{new}})}{P(\mathcal{D}|\mathcal{M}_{\text{old}})P(\mathcal{M}_{\text{old}}|\beta_1,\beta_2)Q(\mathcal{M}_{\text{new}}|\mathcal{M}_{\text{old}})}, 1\right\} \quad (6.25)$$

$$A(\beta_{1\text{new}}|\beta_{1\text{old}}) = \min\left\{\frac{P(\mathcal{M}|\beta_{1\text{new}},\beta_2)}{P(\mathcal{M}|\beta_{1\text{old}},\beta_2)}, 1\right\} \quad (6.26)$$

$$A(\beta_{2\text{new}}|\beta_{2\text{old}}) = \min\left\{\frac{P(\mathcal{M}|\beta_1,\beta_{2\text{new}})}{P(\mathcal{M}|\beta_1,\beta_{2\text{old}})}, 1\right\} \quad (6.27)$$

6.2.2.3 Practical issues

In our simulations, we chose the prior distribution of the hyperparameters $P(\beta)$ to be the uniform distribution over the interval $[0, \text{MAX}]$. The proposal probability for the hyperparameters $R(\beta_{\text{new}}|\beta_{\text{old}})$ was chosen to be a uniform distribution over a moving interval of length $2l \ll \text{MAX}$, centred on the current value of the hyperparameter. Consider a hyperparameter β_{new} to be sampled in an MCMC move given that we have the current value β_{old}. The proposal distribution is uniform over the interval $[\beta_{\text{old}}-l, \beta_{\text{old}}+l]$ with the constraint that $\beta_{\text{new}} \in [0, \text{MAX}]$. If the sampled value β_{new} happens to lie outside the allowed interval, the value is

reflected back into the interval. The respective proposal probabilities can be shown to be symmetric (see appendix A) and therefore to cancel out in the acceptance probability ratio. In our simulations, we set the upper limit of the prior distribution to be MAX = 30, and the length of the sampling interval to be $l = 3$. Note that the choice of l only affects the convergence and mixing of the Markov chain, but has theoretically no influence on the results. While an adaptation of this parameter during burn-in could be attempted to optimize the computational efficiency of the scheme, we found that the chosen value of l gave already a fast convergence of the Markov chain that we did not deem necessary to further improve.

To test for convergence of the MCMC simulations, various methods have been developed; see Cowles and Carlin (1996) for a review. In our work, we applied the simple scheme used in Friedman and Koller (2003): each MCMC run was repeated from independent initializations, and consistency in the marginal posterior probabilities of the edges was taken as indication of sufficient convergence. This approach is discussed in more detail in Section 3.2.2.4. For the applications reported in Section 6.5, this led to the decision to run the MCMC simulations for a total number of 5×10^5 steps, of which the first half were discarded as the burn-in phase.

6.3 Simulations

The objective of this section is to explore the posterior probability landscape in the space of hyperparameters. This will help us to better interpret the values of the hyperparameters sampled with MCMC in real applications, and to assess whether these values are plausible. We pursue this objective with two different approaches. In the first approach, we design a hypothetical population of network structures for which we can analytically derive a closed-form expression of

the partition function and, hence, the marginal posterior probability of the hyperparameters. These results will be presented in Subsections 6.3.1 and 6.3.3 for one and multiple sources of prior knowledge, respectively. In the second approach, we focus on a small network with a limited number of nodes. Although we cannot derive a closed-form expression for the partition function in this case, we can compute the partition function numerically via an exhaustive enumeration of all possible network structures; this again allows us to compute the marginal posterior probability of the hyperparameters. The resulting posterior probability landscapes will be presented in Subsections 6.3.2 and 6.3.4, again for one and multiple sources of prior knowledge, respectively. We compare these results with the values of hyperparameters sampled from an MCMC simulation; this approximate numerical procedure is the only approach that is viable in real-world applications with many interacting nodes.

6.3.1 Idealized derivation for one source of prior knowledge

Consider the partition of a hypothetical space of network structures, depicted in Figure 6.2. This Venn diagram consists of four mutually exclusive subsets, which represent networks that are characterized by different compatibilities with respect to the data and the prior knowledge. We make the idealizing assumption that the networks either completely succeed or fail in modelling the data. The networks are also assumed to be either completely consistent or inconsistent with the assumed prior knowledge. The different sizes of the subsets are related to the relative proportions of the networks they contain, which are described by the following quantities:

- **TD**: Proportion of networks that are in agreement with the data only.
- **TD1**: Proportion of networks that are in agreement with the data and with the prior.

Graph in agreement with:		Result		
Data	Prior	$P(\mathcal{D}\|\mathcal{M})$	E	Proportion
no	no	a	1	F
no	yes	a	0	T1
yes	no	A	1	TD
yes	yes	A	0	TD1

Table 6.1: **Idealized scenario for one source of prior.** This table summarizes the definitions for the idealized population of network structures when considering one source of biological prior knowledge, corresponding to the Venn diagram of Figure 6.2.

- **T1**: Proportion of networks that are in agreement with the prior only.
- **F**: Proportion of networks that are neither in agreement with the data nor with the prior.

We define that networks that are in agreement with the data have marginal likelihood $P(\mathcal{D}|\mathcal{M}) = A$, while those in disagreement with the data have the lower marginal likelihood $P(\mathcal{D}|\mathcal{M}) = a$, with $a < A$. In our experiments discussed below, we set $A = 10$ and $a = 1$. A network that is in accordance with the biological prior knowledge has zero energy $E = 0$; otherwise, the network is penalized with a higher energy of $E = 1$. Table 6.1 presents a summary of these definitions. We want to find the posterior distribution $P(\beta|\mathcal{D})$:

$$P(\beta|\mathcal{D}) \;=\; \frac{1}{P(\mathcal{D})} \sum_{\mathcal{M}} P(\mathcal{D}, \mathcal{M}, \beta) \tag{6.28}$$

The conditional independence relations, represented by the graphical model in the left panel of Figure 6.1, imply that

$$P(\mathcal{D}, \mathcal{M}, \beta) \;=\; P(\mathcal{D}|\mathcal{M}) P(\mathcal{M}|\beta) P(\beta) \tag{6.29}$$

Assuming a uniform prior over β, we thus obtain

$$P(\beta|\mathcal{D}) \;\propto\; \sum_{\mathcal{M}} P(\mathcal{D}|\mathcal{M}) P(\mathcal{M}|\beta) \tag{6.30}$$

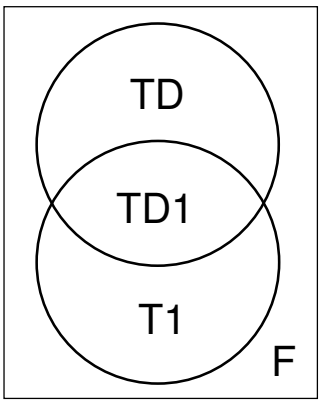


Figure 6.2: **Venn diagram for an idealized population of network structures and one source of prior knowledge.** The Venn diagram shows a hypothetical population of network structures. We make the idealizing assumption that the networks either completely succeed or fail in modelling the data. The networks are also assumed to be either completely consistent or inconsistent with the assumed prior knowledge. TD is the proportion of graphs that agree with the data. TD1 is the proportion of graphs that agree with the data and the biological prior knowledge. T1 is the proportion of graphs that agree with the biological prior knowledge only. F is the proportion of graphs that are neither in agreement with the data nor with the biological prior knowledge. A summary of this scenario is provided in Table 6.1.

Inserting the expression for the prior distribution, Equations 6.2-6.3, into this sum, we get:

$$\sum_{\mathcal{M}} P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\beta) \;=\; \frac{\sum_{\mathcal{M}} P(\mathcal{D}|\mathcal{M})e^{-\beta E(\mathcal{M})}}{\sum_{\mathcal{M}} e^{-\beta E(\mathcal{M})}} \tag{6.31}$$

Using the definitions from Table 6.1, we thus obtain the following expression for the posterior distribution $P(\beta|\mathcal{D})$:

$$P(\beta|\mathcal{D}) \;\propto\; \frac{a \times T1 + A \times TD1 + e^{-\beta}(a \times F + A \times TD)}{TD1 + T1 + e^{-\beta}(F + TD)} \tag{6.32}$$

where we refer to the expression on the right as the unnormalized posterior distribution. A plot of this distribution is shown in the left panel of Figure 6.5.

6.3.2 Simulation results for one source of prior knowledge

The objective of this subsection is to compare the closed form of the posterior distribution $P(\beta|\mathcal{D})$ from Equation 6.32 with that obtained from a synthetic study using real Bayesian networks. To this end, we consider a Bayesian network with

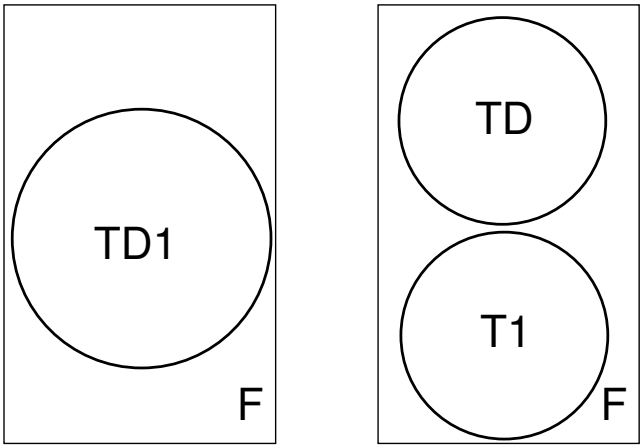


Figure 6.3: **Venn diagrams for a completely correct and a completely wrong source of biological prior knowledge.** The two Venn diagrams show special scenarios of the hypothetical network population depicted in Figure 6.2. The left panel represents the situation of completely correct prior knowledge. All networks that are consistent with the data also agree with the prior, and all networks that are in accordance with the prior also agree with the data. Hence $T1 = TD = 0$. The right panel shows the situation of a completely wrong source of prior knowledge. Networks that are consistent with the data are not supported by the prior, while networks that are in agreement with the prior contradict the findings in the data. Hence $TD1 = 0$. (For a definition of the symbols, see Table 6.1 and the caption of Figure 6.2).

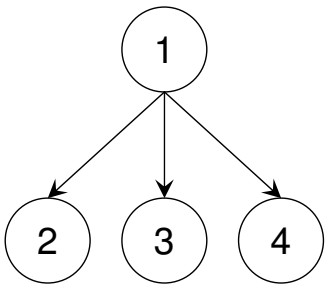


Figure 6.4: **HUB network.** This figure shows the network structure from which we generated data for the synthetic inference study.

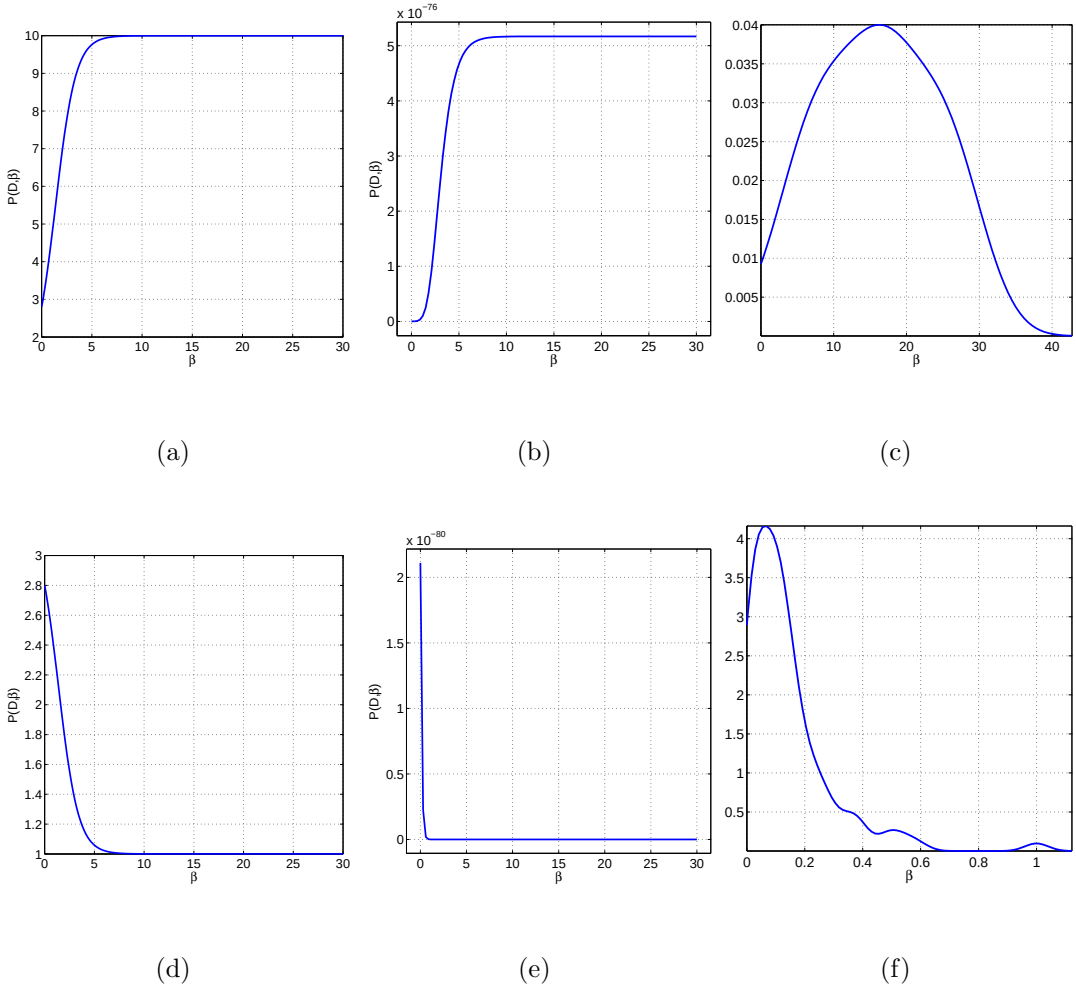


Figure 6.5: **Results of the simulation study for a single source of prior knowledge.** The top row shows the results when including the correct prior knowledge. The bottom row shows the results when the prior knowledge is wrong. The left column shows the unnormalized posterior probability of the hyperparameter β for the idealized network population depicted in Figure 6.3, computed from Equation 6.32 and plotted against β. The values of the network population proportions, defined in Table 6.1 and Figure 6.2, were set as follows. Correct prior (corresponding to the left panel in Figure 6.3): $TD = T1 = 0, TD1 = 0.2$. Wrong prior (corresponding to the right panel in Figure 6.3): $TD = T1 = 0.2, TD1 = 0$. The centre column shows the unnormalized posterior probability of β for the synthetic toy problem, plotted against β. For comparison, the right column shows the marginal posterior probability densities of β, estimated from the MCMC trajectories with a Parzen estimator, using a Gaussian kernel whose standard deviation was set automatically by the MATLAB function `ksdensity.m`. The MCMC scheme was discussed in Section 6.2.2.

a small number of nodes such that a complete enumeration of all possible network structures is possible. This allows the partition function in Equation 6.3 and hence the posterior distribution $P(\beta|\mathcal{D})$ to be computed exactly, the latter via Equations 6.2 and 6.30. We consider the two extreme scenarios of completely correct and completely wrong prior knowledge. For the idealized network population, the situation of completely correct prior knowledge is depicted in the Venn diagram on the left of Figure 6.3: all networks that accord with the prior also accord with the data, while networks not according with the prior also fail to accord with the data. The Venn diagram on the right of Figure 6.3 depicts the opposite scenario of completely wrong prior knowledge: networks that accord with the data never accord with the prior while, conversely, networks that accord with the prior never accord with the data. For the synthetic toy problem, the completely correct prior corresponds to a prior knowledge matrix B that is identical to the true adjacency matrix $\mathcal{M}$ of the network (see Section 6.2.1 for a reminder of this terminology). On the contrary, completely wrong prior knowledge corresponds to a prior knowledge matrix B that is the complete complement of the network adjacency matrix $\mathcal{M}$, that is, has entries indicating edges where there are none in the true network and, conversely, has zero entries for the locations of the true edges in the network.

The network that we used for the synthetic toy problem is shown in Figure 6.4. We treated it as a DBN and generated a time series of 100 exemplars from it, as described in Section 6.4.1. The results are shown in Figure 6.5, where the top row corresponds to the true prior, and the bottom row to the wrong prior. The left and centre columns show plots of the (unnormalized) posterior distribution of the hyperparameter β for the idealized network population and the synthetic toy problem, respectively. The graphs are similar, as expected. In both cases, when the prior is correct, $P(\beta|\mathcal{D})$ monotonically increases until it reaches a plateau. When the prior is wrong, $P(\beta|\mathcal{D})$ peaks at zero, and monotonically

decreases for increasing values of β. For comparison, the right column shows the marginal posterior probability densities of β estimated from the MCMC trajectories. The MCMC scheme was discussed in Section 6.2.2. All results are consistent in indicating that for the true prior, high values of β are encouraged, while for the wrong prior, high values of β are suppressed. Since β represents the weight that is assigned to the prior, our finding confirms that the proposed methodology is working as expected. It also lays the foundations for investigating the more complex scenario of multiple sources of prior knowledge, to be discussed next.

6.3.3 Idealized derivation for two sources of prior knowledge

Next, we generalize the scenario of Subsection 6.3.1 to two independent sources of prior knowledge. Again, consider a hypothetical space of network structures, which is assumed to be partitioned into distinct regions, as depicted by the Venn diagram of Figure 6.6. The symbols in this diagram indicate the proportions of networks that fall into the respective regions:

- **TD** is the proportion of graphs that are in agreement with the data only.
- **TD1** is the proportion of graphs that are in agreement with the data and with the first source of prior knowledge.
- **T1** is the proportion of graphs that are in agreement with the first source of prior knowledge only.
- **T2** is the proportion of graphs that are in agreement with the second source of prior knowledge only.
- **TD2** is the proportion of graphs that are in agreement with the data and with the second source of prior knowledge.
- **TD12** is the proportion of graphs that are in agreement with the data and with both sources of prior knowledge.

Graph in agreement with:			Result			
Data	Prior 1	Prior 2	$P(\mathcal{D}\|\mathcal{M})$	E_1	E_2	Proportion
no	no	no	a	1	1	F
no	no	yes	a	1	0	T2
no	yes	no	a	0	1	T1
no	yes	yes	a	0	0	T12
yes	no	no	A	1	1	TD
yes	no	yes	A	1	0	TD2
yes	yes	no	A	0	1	TD1
yes	yes	yes	A	0	0	TD12

Table 6.2: **Idealized scenario for two independent sources of prior knowledge.** This table summarizes the definitions for the idealized population of network structures with two sources of prior knowledge, corresponding to Figure 6.6.

- **T12** is the proportion of graphs that are in agreement with both sources of prior knowledge, but not the data.

- **F** is the proportion of graphs that are neither in agreement with the data, nor with any prior.

We define that networks that are in agreement with the data have marginal likelihood $P(\mathcal{D}|\mathcal{M}) = A$, while networks not in agreement with the data have the lower marginal likelihood $P(\mathcal{D}|\mathcal{M}) = a$, with $a < A$. In our experiments we set $A = 10$ and $a = 1$. Networks that are in accordance with the first source of prior knowledge have energy $E_1 = 0$, otherwise the energy is $E_1 = 1$. Networks that are in accordance with the second source of prior knowledge have energy $E_2 = 0$, otherwise the energy is $E_2 = 1$. Table 6.2 presents a summary of these definitions. Generalizing the derivation presented in Subsection 6.3.1, we now want to find the posterior distribution of both hyperparameters $P(\beta_1, \beta_2|\mathcal{D})$:

$$P(\beta_1, \beta_2|\mathcal{D}) \;=\; \frac{1}{P(\mathcal{D})} \sum_{\mathcal{M}} P(\beta_1, \beta_2, \mathcal{D}, \mathcal{M}) \tag{6.33}$$

From the conditional independence relations depicted by the graphical model in the right panel of Figure 6.1, we get:

$$P(\mathcal{D}, \mathcal{M}, \beta_1, \beta_2) \;=\; P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\beta_1, \beta_2)P_1(\beta_1)P_2(\beta_2) \tag{6.34}$$

Assuming uniform priors over the two hyperparameters β_1 and β_2, we obtain:

$$P(\beta_1, \beta_2|\mathcal{D}) \;\propto\; \sum_{\mathcal{M}} P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\beta_1, \beta_2) \tag{6.35}$$

Inserting the expression for the prior, Equations 6.9-6.10, into this sum, we get:

$$\sum_{\mathcal{M}} P(\mathcal{D}|\mathcal{M})P(\mathcal{M}|\beta_1, \beta_2) = \frac{\sum_{\mathcal{M}} P(\mathcal{D}|\mathcal{M}) e^{[-\beta_1 E_1(\mathcal{M}) - \beta_2 E_2(\mathcal{M})]}}{\sum_{\mathcal{M}} e^{[-\beta_1 E_1(\mathcal{M}) - \beta_2 E_2(\mathcal{M})]}} \tag{6.36}$$

Using the definitions from Table 6.2, this yields:

$$P(\beta_1, \beta_2|\mathcal{D}) \propto \frac{e^{-\beta_2}(a[T1] + A[TD1]) + e^{-\beta_1}(a[T2] + A[TD2]) + \ldots}{e^{-\beta_2}(T1 + TD1) + e^{-\beta_1}(T2 + TD2) + \ldots} \\ \frac{\ldots + e^{(-\beta_1 - \beta_2)}(a[F] + A[TD]) + a[T12] + A[TD12]}{\ldots + e^{(-\beta_1 - \beta_2)}(TD + F) + TD12 + T12} \tag{6.37}$$

where, again, we refer to the expression on the right as the unnormalized posterior distribution of the hyperparameters. A plot of this distribution is shown in the top left panel of Figure 6.8.

6.3.4 Simulation results for two sources of prior knowledge

We revisit the simulations discussed in Subsection 6.3.2, where we have considered two sources of prior knowledge, one being correct and the other being completely wrong. Rather than studying the effects of these priors in isolation, we now combine them and integrate them simultaneously into the inference scheme. For the idealized population of network structures, the situation is illustrated in Figure 6.7. The posterior probability distribution of the two hyperparameters is computed from Equation 6.37, using the parameter setting stated in the captions of Figures 6.7 and 6.8. For the synthetic toy problem, the prior probability

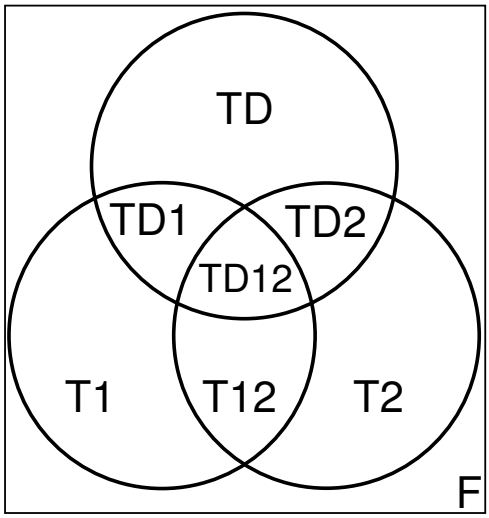


Figure 6.6: **Venn diagram for an idealized population of network structures and multiple sources of prior knowledge.** This Venn diagram is a generalization of Figure 6.2 for two independent sources of prior knowledge. TD is the proportion of networks that agree with the data. TD1 is the proportion of networks that agree with the data and prior 1. T1 is the proportion of networks that agree with prior 1 only. TD2 is the proportion of networks that agree with the data and prior 2. T2 is the proportion of networks that agree with prior 2 only. TD12 is the proportion of networks that agree with the data and with both priors. T12 is the proportion of networks that agree with both priors but not the data. F is the proportion of networks that are neither in agreement with the data nor the biological prior knowledge. A summary of this scenario can be found in Table 6.2.

distribution over network structures is computed from Equation 6.9, obtaining the partition function of Equation 6.10 from a complete enumeration of all possible network structures. The posterior distribution of the hyperparameters is then computed from Equation 6.35, again resorting to a complete enumeration of network structures. For comparison, we also sampled the hyperparameters from the posterior distribution numerically, using the MCMC scheme described in Section 6.2.2.2. The results are shown in Figure 6.8. The bottom left panel shows the trace plots from the MCMC simulation. The values of β_2, the hyperparameter associated with the wrong prior, are always below those of β_1, the hyperparameter associated with the true prior. This confirms our expectation that the inference scheme succeeds in distinguishing between the different priors and automatically associates a higher weight with the correct prior. Somewhat counterintuitively, though, the value of β_2 does not decay to zero, suggesting that the second prior,

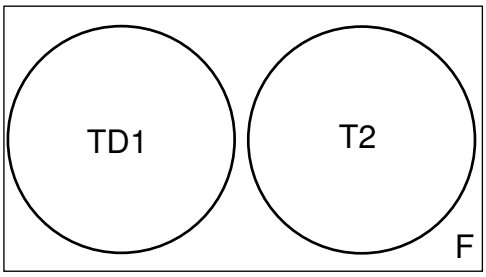


Figure 6.7: **Venn diagram for a completely correct and a completely wrong source of biological prior knowledge.** This Venn diagram shows a special case of Figure 6.6 where one source of biological prior knowledge is in complete agreement with the data while the other source of prior knowledge is completely wrong. All networks that are consistent with the data also accord with the first prior, and all networks that are in accordance with the first prior also agree with the data. Hence $T1 = TD = 0$. Networks that are consistent with the data are not supported by the second prior, while networks that are in agreement with the second prior contradict the findings in the data. Hence $TD2 = TD12 = 0$. The priors are also mutually exclusive: $T12 = 0$. Note that the scenario depicted here effectively combines the two scenarios of Figure 6.3. See Table 6.2 and the caption of Figure 6.6 for a definition of the symbols.

despite the worst-case scenario of it being completely wrong, is never 'switched off' completely. This seemingly strange behaviour was also consistently found in our MCMC simulations on the real data – see the discussion in Section 6.5.2.2 – and provided the motivation for the synthetic simulation study discussed in the present section. An elucidation of this behaviour is obtained from the plots of the posterior distribution $P(\beta_1, \beta_2|\mathcal{D})$ in the left and right top panels of Figure 6.8. Both graphs indicate that $P(\beta_1, \beta_2|\mathcal{D})$ contains a ridge parallel to the line $\beta_1 = \beta_2$, dropping to zero for $\beta_1 < \beta_2$, and reaching a plateau for $\beta_1 > \beta_2$. This plateau explains the results found in our MCMC simulations. When β_1 is sufficiently larger than β_2, corresponding to a configuration on the plateau well over the ridge, there is no effective force pushing β_2 down to zero. The intuitive explanation is that for β_1 sufficiently larger than β_2, the effect of the second (wrong) prior is already negligible, so that it becomes obsolete to completely switch it off.

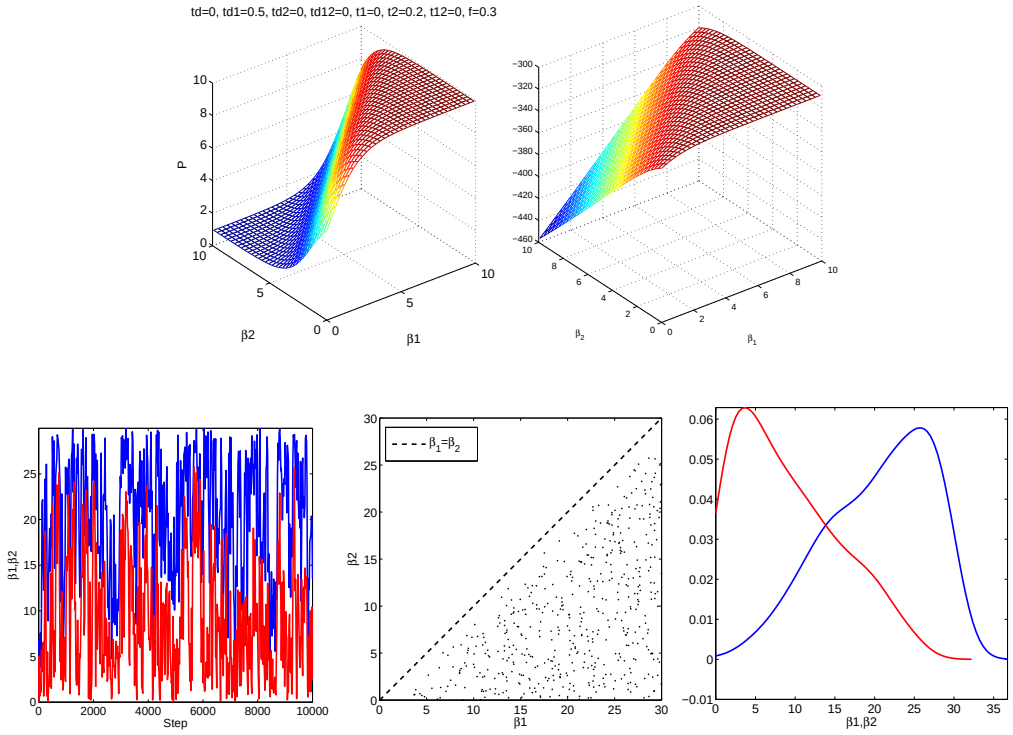


Figure 6.8: **Results of the simulation study for multiple sources of prior knowledge.** This figure shows the inference results for two independent sources of prior knowledge, associated with separate hyperparameters β_1 and β_2. The top left panel shows a plot of the unnormalized posterior probability distribution of β_1 and β_2 for the idealized population of network structures depicted in Figure 6.7. The expression was computed from Equation 6.37 with the following parameter settings: $TD1 = 0.5, T2 = 0.2, F = 0.3$, $TD = TD2 = TD12 = T1 = T12 = 0$ (see the caption of Figure 6.7 for an explanation of why the parameters were chosen in that way). The top right panel shows a plot of the unnormalized posterior distribution of β_1 and β_2 for the synthetic toy problem. The bottom left panel shows two trace plots obtained when sampling the two hyperparameters from the posterior distribution with the MCMC scheme discussed in Section 6.2.2.2. The horizontal axis represents the MCMC step while the vertical axis shows the sampled value of the hyperparameter. The bottom central panel shows a scatter plot of $\beta_1 \times \beta_2$ in order to make it clear that $\beta_1 > \beta_2$. The bottom right panel shows the marginal posterior probability densities of β_1 and β_2, estimated from the MCMC trajectories with a Parzen estimator, using a Gaussian kernel whose standard deviation was set automatically by the MATLAB function `ksdensity.m`. The blue graph corresponds to β_1, the hyperparameter associated with the true prior. The red graph corresponds to β_2, the hyperparameter associated with the wrong prior.

6.4 Data and priors

6.4.1 Simulated data

The data generated for the synthetic simulations described in Section 6.3 were obtained from a DBN with a linear Gaussian distribution. See Section 3.3 for a more detailed discussion about dynamic Bayesian networks.

The random variable $X_i(t+1)$ denoting the expression of node i at time $t+1$ is distributed according to:

$$X_i(t+1) \sim N\left(\sum\nolimits_k w_{ik}x_k(t), \sigma^2\right) \tag{6.38}$$

where $N(.)$ denotes the Normal distribution, the sum extends over all parents of node i, and $x_k(t)$ represents the value of node k at time t. We set the standard deviation to $\sigma = 0.1$, and the interaction strengths to $w_{ik} = 1$. The structure of the network from which we generated data is represented in Figure 6.4.

6.4.2 Yeast cell cycle

For the evaluation of the proposed inference method, we were guided by the study of Bernard and Hartemink (2005). The authors aimed to infer regulatory networks involving 25 genes of yeast (*Saccharomyces cerevisiae*), of which 10 genes encode known transcription factors (TFs). The inference was based on gene expression data, combined with prior knowledge about transcription factor binding locations. The gene expression data were obtained from Spellman et al. (1998); this data set contains 73 time points collected over 8 cycles of the yeast cell cycle using four different synchronization protocols. The prior knowledge about transcription factor binding locations was obtained from the chromatin immunoprecipitation (ChIP-on-chip) assays of Lee et al. (2002).

In our study, we followed the approach of Bernard and Hartemink (2005), but complemented their evaluation by the inclusion of additional gene expression data

and a separate source of prior knowledge. As further gene expression data we included the results of microarray experiments carried out by Tu et al. (2005); this data set contains 36 time points of gene expression data in yeast, collected over three consecutive metabolic cycles in intervals of 25 minutes. As additional prior knowledge, we included the TF binding locations obtained from an independent chromatin immunoprecipitation assay, reported in Harbison et al. (2004). In order to include these binding locations in the proposed inference scheme, we transformed the p-values obtained from the immunoprecipitation assays into probabilities, using the transformation proposed by Bernard and Hartemink (2005). The distribution of p-values is assumed to be exponential if the edge is present and to be uniform if the edge is not present. With these definitions and applying Bayes rule Bernard and Hartemink (2005) shows that the probability of an edge being present ($\mathcal{M}_{ij} = 1$) after a p-value is observed ($\rho_i = p$) is:

$$P_\lambda(\mathcal{M}_{ij} = 1|\rho_i = p) = \frac{\lambda e^{-\lambda p}\zeta}{\lambda e^{-\lambda p}\zeta + (1 - e^{-\lambda})(1 - \zeta)} \tag{6.39}$$

where ζ is the the probability that the edge is present before the p-value is observed. The parameter that controls the scale of the truncated exponential distribution is λ. Instead of setting one value for λ it is assumed to be uniformily distributed on the interval $[\lambda_L, \lambda_H]$ and it is integrated out to yield:

$$P(\mathcal{M}_{ij} = 1|\rho_i = p) = \frac{1}{\lambda_H - \lambda_L} \int_{\lambda_L}^{\lambda_H} \frac{\lambda e^{-\lambda p}\zeta}{\lambda e^{-\lambda p}\zeta + (1 - e^{-\lambda})(1 - \zeta)} d\lambda \tag{6.40}$$

For a detailed discussion about these transformation see Bernard and Hartemink (2005).

The probabilities obtained with this transformation formed the entries B_{ij} of our biological prior knowledge matrix. However, only 10 of the 25 studied genes are known to be TFs. For the remaining genes, no information about binding locations is available. The respective entries in the prior knowledge matrix were thus set to $B_{ij} = 0.5$, corresponding to the absence of prior information (see the discussion in Section 6.2.1).

Summarizing, we evaluated the performance of the proposed inference scheme on two sets of gene expression data and two sets of TF binding location indications. An overview is given in Table 6.3.

6.4.3 Raf signalling pathway

The flow cytometry data (Sachs et al., 2005) used in this is study is presented in Section 4.2. In this set of experiments we use only the observational data leaving the interventional data out. We use the same 5 data sets with 100 measurements each that were presented in Section 4.2.

We extracted biological prior knowledge from the Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways database (Kanehisa, 1997; Kanehisa and Goto, 2000; Kanehisa et al., 2006). KEGG pathways represent current knowledge of the molecular interaction and reaction networks related to metabolism, other cellular processes, and human diseases. As KEGG contains different pathways for different diseases, molecular interactions and types of metabolism, it is possible to find the same pair of genes[1] in more than one pathway. We therefore extracted all pathways from KEGG that contained at least one pair of the 11 proteins/phospholipids included in the Raf pathway. We found 20 pathways, including metabolic and signalling, that satisfied this condition. From these pathways, we computed the prior knowledge matrix, introduced in Section 6.2.1, as follows. Define by M_{ij} the total number of times a pair of genes i and j appears in a pathway, and by m_{ij} the number of times the genes are connected by a (directed) edge in the KEGG pathway. The elements B_{ij} of the prior knowledge matrix are then defined by

$$B_{ij} = \frac{m_{ij}}{M_{ij}} \tag{6.41}$$

If a pair of genes is not found in any of the KEGG pathways, we set the respective

[1] We use the term "gene" generically for all interacting nodes in the network. This may include proteins encoded by the respective genes.

	Expression Data	1st source of Prior	2nd source of Prior
1	Spellman	Lee	Harbison
2	Tu	Lee	Harbison
3	Spellman	Lee	MCMC Tu
4	Tu	Lee	MCMC Spellman

Table 6.3: **Yeast evaluation settings.** This table summarizes the evaluation procedures we used on the yeast data. The table shows the name of the first author of the data sets that we used. Gene expression data: Spellman et al. (1998) and Tu et al. (2005). TF binding location assays: Lee et al. (2002) and Harbison et al. (2004). The entries *MCMC Spellman* and *MCMC Tu* indicate that the prior knowledge matrix was composed of the marginal posterior probabilities of directed pairwise gene interactions (edges) obtained from running MCMC simulations without prior knowledge on the respective expression data set.

prior association to $B_{ij} = 0.5$, implying that we have no information about this relationship.

6.5 Results

6.5.1 Yeast cell cycle

For evaluating the performance of the proposed Bayesian inference scheme on the yeast cell cycle data, we followed Bernard and Hartemink (2005) with the extension described in Section 6.4.2. We associated the edges of the BN with conditional probabilities of the multinomial distribution family. In this case, the marginal likelihood $P(\mathcal{D}|\mathcal{M})$ of Equation 3.7 on Section 3.2.2 is given by the so-called BDe score; see Heckerman (1999) and Section 3.2.5.1 for details. The chosen form of conditional probabilities requires a discretization of the data. Like Bernard and Hartemink (2005), we discretized the gene expression data into three levels using the information bottleneck algorithm, proposed by Hartemink (2001). We represented information about the cell cycle phase with a separate node, which was forced to be a root node connected to all the nodes in the domain. In all

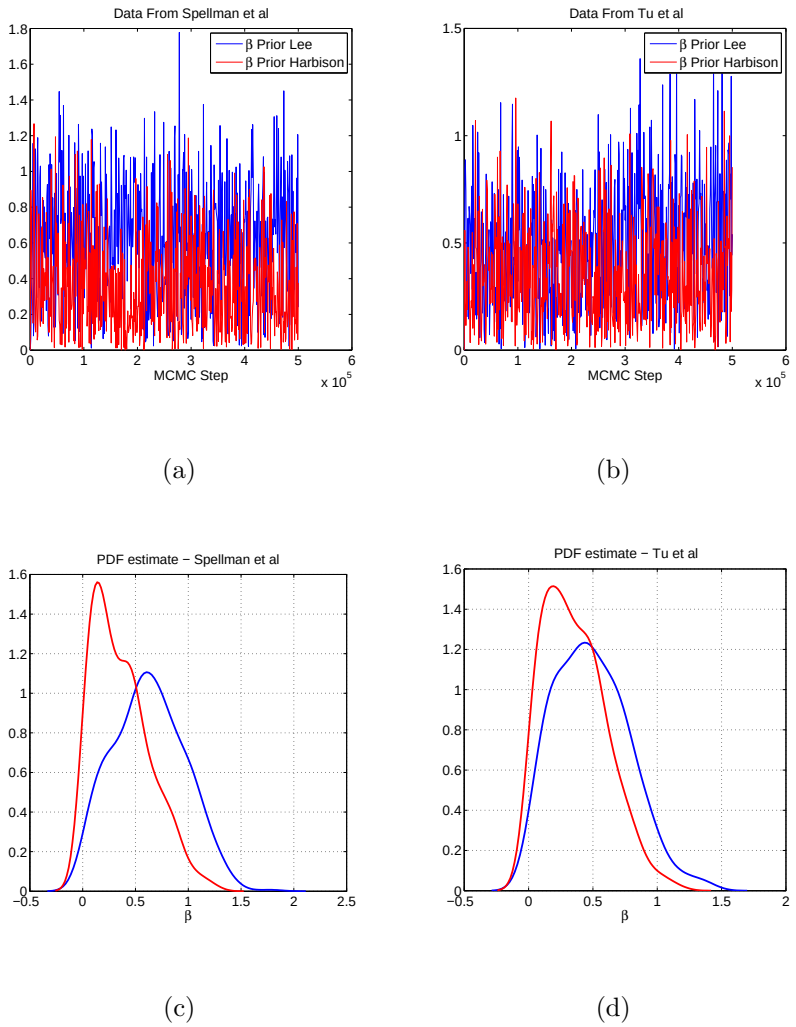


Figure 6.9: **Inferring hyperparameters associated with TF binding locations from gene expression data of yeast.** The top row (a,b) shows the hyperparameter trajectories for two different sources of prior knowledge, sampled from the posterior distribution with the MCMC scheme discussed in Section 6.2.2.2. The bottom row (c,d) shows the corresponding marginal posterior probability densities, estimated from the MCMC trajectories with a Parzen estimator, using a Gaussian kernel whose standard deviation was set automatically by the MATLAB function `ksdensity.m`. The blue line represents the hyperparameter associated with the TF binding locations of Lee et al. (2002). The red line shows the hyperparameter associated with the TF binding locations of Harbison et al. (2004). The two columns are related to different yeast microarray data. Left column: Spellman et al. (1998). Right column: Tu et al. (2005). The two experiments correspond to the first two rows of Table 6.3.

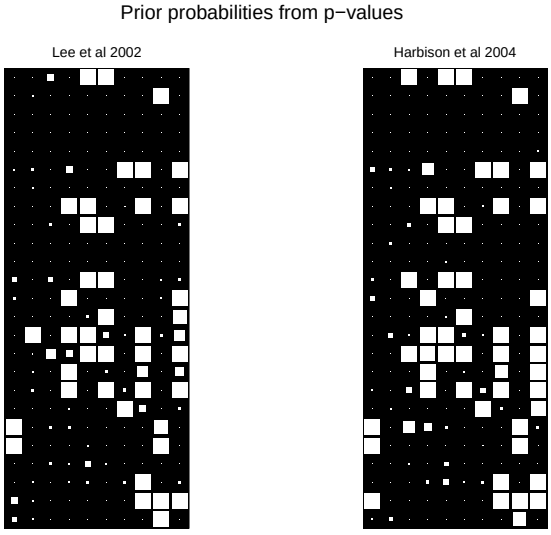


Figure 6.10: **Transcription factor (TF) binding locations.** The two Hinton diagrams provide a qualitative display of the TF binding location assays of Lee et al. (2002) (left panel) and Harbison et al. (2004) (right panel). The columns of the two matrices represent 10 known TFs. The rows represent 25 genes that are putatively regulated by the TFs. The size of a white square represents the probability that a TF binds to the promoter of the respective gene, with a larger square indicating a value closer to 1. These probabilities were obtained by subjecting the p-values from the original immunopreciptation experiments of Lee et al. (2002) and Harbison et al. (2004) to the transformation proposed by Bernard and Hartemink (2005).

our MCMC simulations, we combined gene expression data with two independent sources of prior knowledge, and sampled networks and hyperparameters from the conditional probability distribution according to the MCMC scheme described in Section 6.2.2.2.

Table 6.3 presents a summary of the simulation settings we used. In our first application, corresponding to the first row of Table 6.3, the gene expression data were taken from Spellman et al. (1998). In our second application, corresponding to the second row of Table 6.3, the gene expression data came from Tu et al. (2005). In both applications, we used the same two independent sources of prior knowledge in the form of transcription factor (TF) binding locations (Lee et al., 2002; Harbison et al., 2004), as described in Section 6.4.2.

The MCMC trajectories of the hyperparameters associated with the two sources of biological prior knowledge are presented in Figure 6.9. The figure

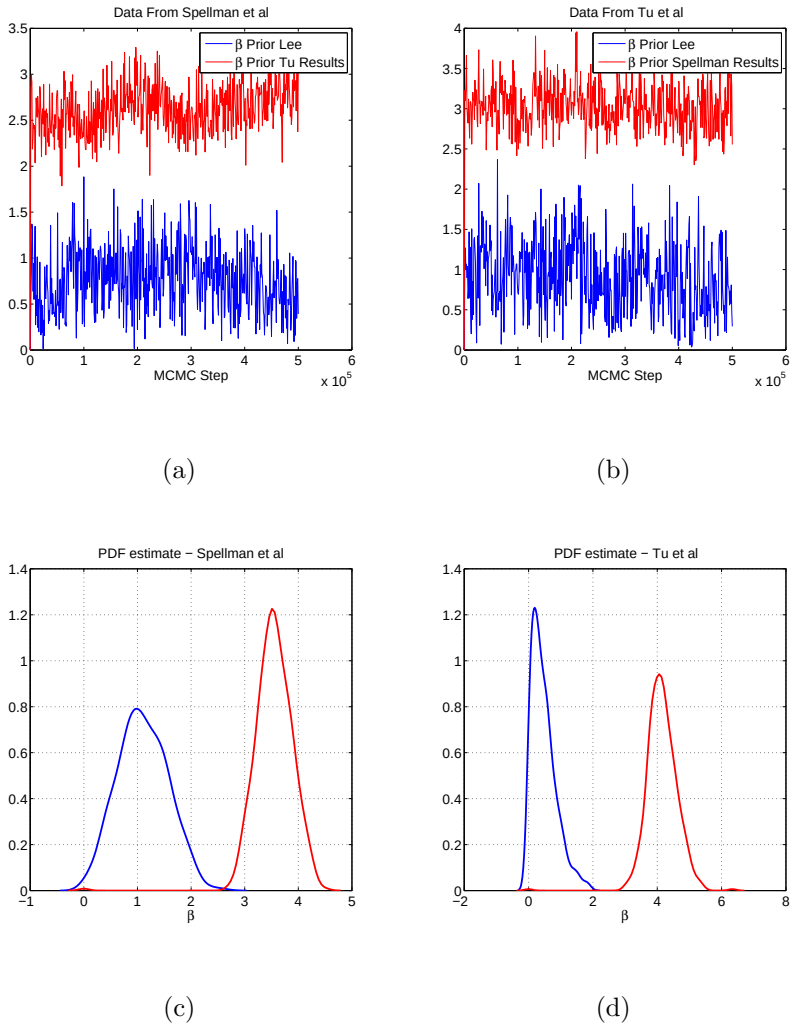


Figure 6.11: **Inferring hyperparameters associated with priors of different nature.** The graphs are similar to those of Figure 6.9, but were obtained for different sources of prior knowledge. The blue lines show the MCMC trace plots (top row) and estimated marginal posterior probability distributions (bottom row) of the hyperparameter associated with the TF binding locations from Lee et al. (2002). The red lines correspond to the hyperparameter associated with prior knowledge obtained from an independent microarray experiment in the way described in Section 6.5.1. The left column shows the results obtained from the experiment corresponding to the third row of Table 6.3. The right column shows the results obtained from the experiment corresponding to the fourth row of Table 6.3. For an explanation of the graphs, see the caption of Figure 6.9.

also shows the estimated marginal posterior probability distributions of the two hyperparameters. These distributions, as well as the MCMC trace plots, do not appear to be very different, which suggests that the two priors are similar. A closer inspection of the results from the two TF binding assays, shown in Figure 6.10, reveals that the indications of putative TF binding locations obtained independently by Lee et al. (2002) and Harbison et al. (2004) are, in fact, very similar. This finding confirms that the results obtained with the proposed Bayesian inference scheme are consistent and in accordance with our expectation. From Figure 6.9 we also note that the sampled values of the hyperparameters are rather small, and that the estimated marginal posterior distributions – compared to those presented in the next section – are quite close to zero. This suggests that the prior information included is not in strong agreement with the data. There are two possible explanations for this effect. First, the TF activities might be controlled by post-translational modifications, which implies that the gene expression data obtained from microarray experiments might not contain sufficient information for inferring regulatory interactions between TFs and the genes they regulate. Second, there might be relevant regulatory interactions between genes that do not belong to the set of a priori known TFs, which are hence inherently undetectable by the binding assays.

One might therefore assume that prior knowledge obtained on the basis of a preceding microarray experiment might be more informative about a subsequent second microarray experiment than TF binding locations. To test this conjecture, we took one of the two gene expression data sets, assumed a uniform prior on network structures (subject to the usual fan-in restriction), and sampled networks from the posterior distribution with MCMC. From this sample, we obtained the marginal posterior probabilities of all edges, and used the resulting matrix as a source of prior knowledge for the subsequent microarray experiment. We proceeded with the settings shown in the third and fourth row of

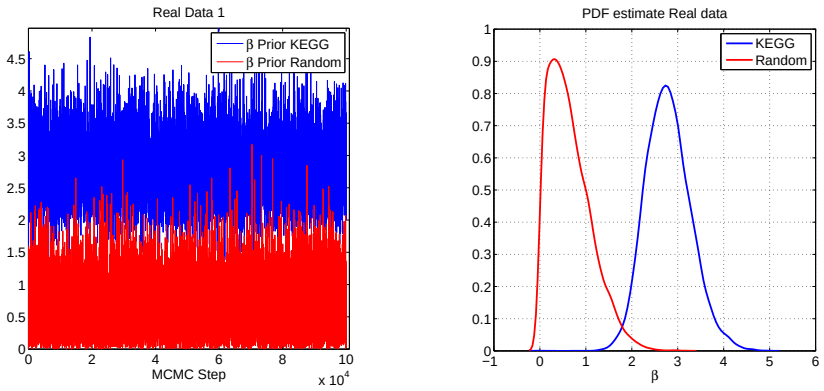
Table 6.3. First, we combined the results obtained from the gene expression data of Spellman et al. (1998) with the binding locations from Lee et al. (2002) and applied these two sources of prior knowledge to the gene expression data from Tu et al. (2005). Second, we combined the results obtained from the gene expression data of Tu et al. (2005) with the binding locations from Lee et al. (2002) and applied these two sources of prior knowledge to the gene expression data from Spellman et al. (1998). The resulting hyperparameter trajectories are presented in Figure 6.11 together with their estimated probability densities. Compared with the previous results of Figure 6.9, there is now a much clearer separation between the two distributions. The sampled values of the hyperparameter associated with the second, independent source of microarray data significantly exceed those of the hyperparameter associated with the binding data. This suggests that prior knowledge that is more consistent with the data is given a stronger weight by the Bayesian inference scheme, in confirmation of our conjecture.

The critical question to ask next is: by how much does the accuracy of network reconstruction improve as a consequence of integrating prior knowledge into the inference scheme? Unfortunately, this evaluation cannot be done for yeast owing to our lack of knowledge about the true gene regulatory interactions and the absence of a proper gold-standard network. To answer this question, we therefore turn to a second application, for which more biological knowledge about the true regulatory processes exists.

6.5.2 Raf signalling pathway

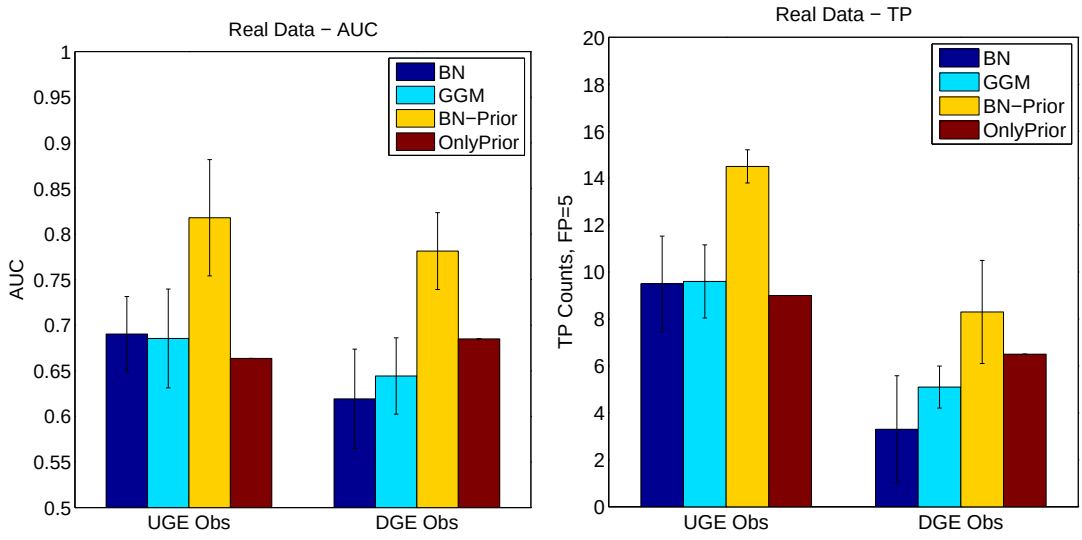
6.5.2.1 Motivation

As described in Section 4.2, the Raf pathway has been extensively studied in the literature. We therefore have a sufficiently reliable gold-standard network for evaluating the results of our inference procedure, as depicted in Figure 4.1.



(a) (b)

Figure 6.12: **Inferring hyperparameters from the cytometry data of the Raf pathway.** The left panel (a) shows the hyperparameter trajectories for two different sources of prior knowledge, sampled from the posterior distribution with the MCMC scheme discussed in Section 6.2.2.2. The right panel (b) shows the corresponding posterior probability densities, estimated from the MCMC trajectories with a Parzen estimator, using a Gaussian kernel whose standard deviation was set automatically by the MATLAB function `ksdensity.m`. The blue lines refer to the hyperparameter associated with the prior knowledge extracted from the KEGG pathways. The red lines refer to completely random and hence vacuous prior knowledge. The data, on which the inference was based, consisted of 100 concentrations of the 11 proteins in the Raf pathway, subsampled from the observational cytometry data of Sachs et al. (2005).



(a) (b)

Figure 6.13: **Reconstruction of the Raf signalling pathway with different machine learning methods.** The figure evaluates the accuracy of inferring the Raf signalling pathway from cytometry data and prior information from KEGG. Two evaluation criteria were used. The left panel shows the results in terms of the area under the ROC curve (AUC scores), while the right panel shows the number of predicted true positive (TP) edges for a fixed number of 5 spurious edges. Each evaluation was carried out twice: with and without taking the edge direction into consideration (UGE: undirected graph evaluation, DGE: directed graph evaluation). Four machine learning methods were compared: Bayesian Networks without prior knowledge (BNs), Graphical Gaussian Models without prior knowledge (GGMs), Bayesian Networks with prior knowledge from KEGG (BN-Prior), and prior knowledge from KEGG only (Only Prior). In the latter case, the elements of the prior knowledge matrix (introduced in Section 6.2.1) were computed from Equation 6.41. The histogram bars represent the mean values obtained by averaging the results over five data sets of 100 protein concentrations each, independently sampled from the observational cytometry data of Sachs et al. (2005). The error bars show the respective standard deviations.

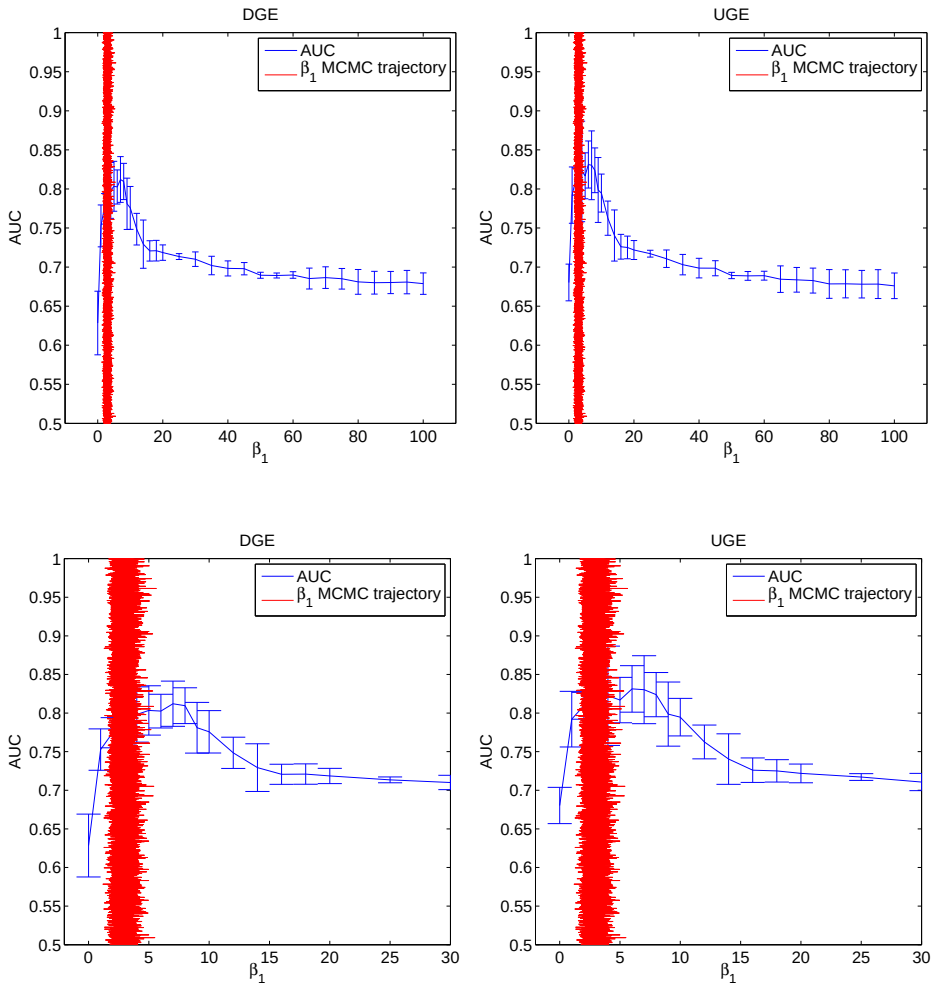


Figure 6.14: **Learning the hyperparameter associated with the prior knowledge from KEGG.** The horizontal axis represents the value of β_1, the hyperparameter associated with the prior knowledge from KEGG. The vertical axis represents the area under the ROC curve (AUC). The blue line shows the mean AUC score for fixed values of β_1, obtained by sampling network structures from the posterior distribution with MCMC. The results were averaged over five data sets of 100 protein concentrations each, independently sampled from the observational cytometry data of Sachs et al. (2005). The error bars show the respective standard deviations. The vertical red lines show trace plots of β_1 obtained with the MCMC scheme described in Section 6.2.2.2, where networks and hyperparameters are sampled from the posterior distribution. Each evaluation was carried out twice, with and without taking the edge direction into consideration. Right panel: undirected graph evaluation (UGE). Left panel: directed graph evaluation (DGE). The bottom row presents a more detailed version of the graphs presented in the top row.

As described in Section 6.4.3, the objective of our study is to assess the viability of the proposed Bayesian inference scheme and to estimate by how much the network reconstruction results improve as a consequence of combining the (down-sampled) cytometry data with prior knowledge from the KEGG pathway database. To this end, we compared the results obtained with the methodology described in Section 6.2 with our earlier results from Werhli et al. (2006) (presented here in Chapter 5), where we had evaluated the performance of Bayesian networks (BNs) and Graphical Gaussian models (GGMs) without the inclusion of prior knowledge. We applied GGMs as described in Schäfer and Strimmer (2005b). For comparability with Werhli et al. (2006), we used BNs with the family of linear Gaussian distributions, for which the marginal likelihood $P(\mathcal{D}|\mathcal{M})$ of Equation 3.7 on Section 3.2.2 is given by the so-called BGe score; see Geiger and Heckerman (1994) and Section 3.2.5.2 for details. Note that the cytometry data of Sachs et al. (2005) are not taken from a time course; hence, BNs were treated as static rather than dynamic models.

6.5.2.2 Discriminating between different priors

We wanted to test whether the proposed Bayesian inference method can discriminate between different sources of prior knowledge and automatically assess their relative merits. To this end, we complemented the prior from the KEGG pathway database with a second prior, for which the entries in the prior knowledge matrix B were chosen completely at random. Hence, this second source of prior knowledge is vacuous and does not include any useful information for reconstructing the regulatory network. Figure 6.12 presents the MCMC trajectories of the hyperparameters β_1 and β_2 together with their respective estimated probability distributions. The hyperparameter associated with the KEGG prior, β_1, takes on substantially larger values than the hyperparameter associated with the vacuous prior, β_2. The estimated posterior distribution of β_1 covers considerably larger

values than the estimated posterior distribution of β_2. This suggests that the proposed method successfully discriminates between the two priors and effectively suppresses the influence of the vacuous prior. Note that the vacuous prior is not completely 'switched off', though, and that the sampled values of β_2 are still substantially larger than zero. This seemingly counterintuitive behaviour is not a failure of the method, but rather an intrinsic feature of the posterior probability landscape; see Figure 6.8 and the discussion in Section 6.3.4.

6.5.2.3 Reconstructing the regulatory network

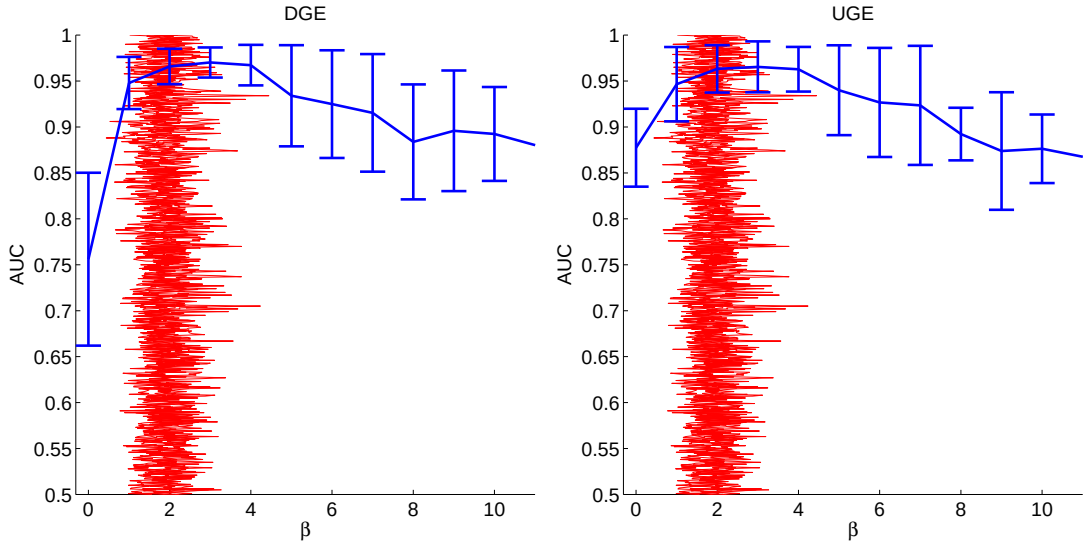
In order to assess the performance of the algorithm in recovering the network we apply two criteria: The AUC and the number of TP for fixed FP=5. Furthermore we consider the directed and undirected graphs namely the DGE and UGE scores respectively. These evaluation criteria are the same that we applied in our comparison study (see Section 5.4) and they are all explained in more detail on Section 4.4.

The results are shown in Figure 6.13. The proposed Bayesian inference scheme clearly outperforms the methods that do not include the prior knowledge from the KEGG database (BNs and GGMs). It also clearly outperforms the prediction that is solely based on the KEGG pathways alone without taking account of the cytometry data. The improvement is significant for all four evaluation criteria: AUC and TP scores for both directed (DGE) and undirected (UGE) graph evaluations. This suggests that the network reconstruction accuracy can be substantially improved by systematically integrating expression data with prior knowledge about pathways, as extracted from the literature or databases like KEGG.

6.5.2.4 Learning the hyperparameters

While the study described in Section 6.5.2.2 suggests that the proposed Bayesian inference scheme succeeds in suppressing irrelevant prior knowledge, we were cu-

rious to see whether the hyperparameter associated with the relevant prior (from KEGG) was optimally inferred. To this end, we chose a large set of fixed values for β_1, while keeping the hyperparameter associated with the vacuous prior fixed at zero: $\beta_2 = 0$. For each fixed value of β_1, we sampled BNs from the posterior distribution with MCMC, and evaluated the network reconstruction accuracy using the evaluation criteria described in Section 6.5.2.3. We compared these results with the proposed Bayesian inference scheme, where both hyperparameters and networks are simultaneously sampled from the posterior distribution with the MCMC scheme discussed in Section 6.2.2.2. The results are shown in Figure 6.14. The blue lines show plots of the various prediction criteria obtained for fixed hyperparameters, plotted against β_1. Plotted along the vertical direction, the red lines show MCMC trace plots for the sampled values of β_1. These results suggest that the inferred values of β_1 are close to those that achieve the best network reconstruction accuracy. However, there is a small yet significant bias: the sampled values of β_1 lie systematically below those that optimize the reconstruction performance. There are two possible explanations for this effect. First, recall that for static BNs as considered here, the partition function of Equation 6.10 is only approximated by Equation 6.16, which could lead to a systematic bias in the inference scheme. Second, it has to be noted that the gold-standard Raf pathway reported in the literature is not guaranteed to be the true biological regulatory network. Recent literature (Dougherty et al., 2005) describes evidence for a negative feedback loop between RAF and ERK via MEK. Active RAF phosphorylates and activates MEK, which, in turn, activates ERK. This corresponds to the directed regulatory path shown in Figure 4.1. However, through a negative feedback mechanism involving ERK, RAF is phosphorylated on inhibitory sites, generating an inactive, desensitized RAF. Details can be found in Dougherty et al. (2005). This feedback loop is not included in the gold-standard network reported in Sachs et al. (2005), shown in Figure 4.1. The existence of a hidden feedback



(a) (b)

Figure 6.15: **Learning the hyperparameter from synthetic data.** The graphs correspond to those of Figure 6.14, but were obtained from five independently generated synthetic data sets. These data were generated from the gold-standard Raf signalling pathway reported in Sachs et al. (2005), as described in Section 6.5.3. The prior knowledge was set to a corrupted version of the gold-standard network, in which 6 (out of the 20) true edges had been removed and replaced by wrong edges. For an explanation of the graphs and symbols, see the caption of Figure 6.14.

loop acting on a putative feedforward path may lead to some systematic error in the edge directions, as static BNs are intrinsically restricted to the modelling of directed acyclic graphs. To shed further light on this issue, we therefore decided to carry out an additional synthetic study.

6.5.3 Comparison with simulated data

We simulated synthetic data from the Raf signalling network, depicted in Figure 4.1 using a linear Gaussian distribution as explained in Section 4.3.1. We set the standard deviation to $\sigma = 0.1$, and the interaction strengths to $w_{ik} = 1$. To mimic the situation described in the previous section, we generated 5 independent data sets with 100 samples each. As prior knowledge, we used a corrupted version of the true network, in which 6 (out of the 20) true edges had been removed and

replaced by wrong edges. We then proceeded with the inference in the same way as described in Section 6.5.2. The results are shown in Figure 6.15, which corresponds to Figure 6.14 for the real cytometry data. From a comparison of these two figures, we note that the small bias in the inference of the hyperparameter has disappeared, and that values of the hyperparameter are sampled in the range where the reconstruction accuracy is optimized. This suggests that the small bias observed in Figure 6.14 might not be caused by the approximation of the partition function in Equation 6.16, but seems more likely to be a consequence of the other two effects discussed at the end of Section 6.5.2 (errors in the gold-standard network and putative feedback loops).

6.6 Modifying the energy function

6.6.1 Introduction

In this section we modify the way a given source of biological prior knowledge is integrated with expression data. In the aforementioned methods the information regarding to the presence and to the absence of edges were treated equally. In this section we use only one source of biological prior knowledge but we split the information contained in it and associate one hyperparameter with the indications about the presence of edges and the other hyperparameter with the indications about the absence of edges.

6.6.2 Methodology

As previously discussed in order to integrate biological prior knowledge into the inference of gene regulatory networks we define a function that measures the agreement between a given network $\mathcal{M}$ and a given source of biological prior knowledge.

Here we use the same ideas that were presented in Section 6.2. We use only one source of prior biological knowledge but we split the energy E in two components. One of the components, E_0, is associated with the absence of edges. The other component, E_1, is associated with the presence of edges. Considering a network $\mathcal{M}$ and a source of prior biological knowledge represented by the matrix B, we define the energies associated with the presence and absence of edges as follows:

$$E_0(\mathcal{M}) = \sum_{\substack{i,j=1 \\ B_{i,j}<0.5}}^{n} |B_{i,j} - \mathcal{M}_{i,j}| \tag{6.42}$$

$$E_1(\mathcal{M}) = \sum_{\substack{i,j=1 \\ B_{i,j}>0.5}}^{n} |B_{i,j} - \mathcal{M}_{i,j}| \tag{6.43}$$

where n is the total number of nodes.

To integrate the prior knowledge expressed by Equations (6.42) and (6.43) into the inference procedure, once again we follow Imoto et al. (2003a) and define the prior distribution over network structures $\mathcal{M}$ to take the form of a Gibbs distribution:

$$P(\mathcal{M}|\beta_0, \beta_1) = \frac{e^{-\{\beta_0 E_0(\mathcal{M}) + \beta_1 E_1(\mathcal{M})\}}}{Z(\beta_0, \beta_1)} \tag{6.44}$$

where the partition function is defined as:

$$Z(\beta_0, \beta_1) = \sum_{\mathcal{M} \in \mathbb{M}} e^{-\{\beta_0 E_0(\mathcal{M}) + \beta_1 E_1(\mathcal{M})\}} \tag{6.45}$$

The two previous equations are in fact the same equations (6.9) and (6.10) respectively. The difference here is not in the equations themselves but in what they represent. While the equations in the previous section represented two different sources of biological prior knowledge being integrated with the expression data here they represent just one source of prior biological knowledge being integrated with the data. The main difference is that in this section one source of prior is split in two, one that indicates the knowledge about the absence of edges, (β_0), and one that indicated the presence of edges, (β_1). The following derivations are also all closely related with the derivations presented in Section 6.2.1.3 on page 111.

As discussed before, unfortunately, the number of graphs increases super-exponentially with the number of nodes, rendering the computation of $Z(\beta_0, \beta_1)$ not viable for large networks. Following the same procedure as in Section 6.2 we define:

$$E_0(\mathcal{M}) = \sum_n \mathcal{E}_0\left(n, \pi_{\mathcal{M}}(n)\right) \tag{6.46}$$

$$E_1(\mathcal{M}) = \sum_n \mathcal{E}_1\left(n, \pi_{\mathcal{M}}(n)\right) \tag{6.47}$$

where $\pi_{\mathcal{M}}(n)$ is the set of parents of node n in the graph $\mathcal{M}$ and we have defined:

$$\mathcal{E}_0\left(n, \pi_{\mathcal{M}}(n)\right) = \sum_{\substack{i \in \pi_{\mathcal{M}}(n) \\ B_{in} < 0.5}} (1 - B_{in}) + \sum_{\substack{i \notin \pi_{\mathcal{M}}(n) \\ B_{in} < 0.5}} B_{in} \tag{6.48}$$

$$\mathcal{E}_1\left(n, \pi_{\mathcal{M}}(n)\right) = \sum_{\substack{i \in \pi_{\mathcal{M}}(n) \\ B_{in} > 0.5}} (1 - B_{in}) + \sum_{\substack{i \notin \pi_{\mathcal{M}}(n) \\ B_{in} > 0.5}} B_{in} \tag{6.49}$$

Following the same rationale presented in Equation (6.16) (page 113) the Equations (6.46) and (6.47) are inserted in Equation (6.45) and thus we obtain:

$$Z(\beta_0, \beta_1) = \prod_n \sum_{\pi_{\mathcal{M}}(n)} e^{-\{\beta_0 \mathcal{E}_0(n, \pi_{\mathcal{M}}(n)) + \beta_1 \mathcal{E}_1(n, \pi_{\mathcal{M}}(n))\}} \tag{6.50}$$

which is the exact partition function for DBNs and an upper bound for static BNs; see the discussion below Equation 6.6.

6.6.3 Simulations

Once again we focus our simulations in the reconstruction of the RAF pathway. The structure of this network is presented in Figure 4.1. Networks and hyper-parameters are sampled with an MCMC sampler according to the methodology presented in Section 6.2.2.2.

In this section observational data from the flow cytometry experiments is combined with a source of prior biological information obtained from the KEGG

database. The data and the source of prior biological knowledge are discussed in more detail in Section 6.4.3. Furthermore, data simulated with Netbuilder is also used in conjunction with the source of prior biological knowledge obtained from KEGG. For details about the Netbuilder simulated data sets see Section 4.3.2.

6.6.4 Results from the modified energy function

Figure 6.16 shows the ROC curves for four different network reconstruction methods: using the prior knowledge from KEGG only, according to Equation (6.41); learning Bayesian networks and graphical Gaussian models from the protein concentration data alone; and the proposed Bayesian inference scheme for integrating prior knowledge and data. The figure also distinguishes between learning the skeleton of the graph only (UGE: undirected graph evaluation) and considering the direction of the edges also (DGE: directed graph evaluation). Recall that larger areas under the ROC curves indicate a better prediction performance overall, although the slope on the left is also of interest, as we are usually interested in keeping the number of false positives bounded at low values. The figure suggests that the systematic integration of prior knowledge with the proposed Bayesian inference scheme leads, overall, to a systematic improvement in the prediction performance over the three alternative schemes that are based on either the data or the prior knowledge from KEGG alone. There are various interesting trends to be noted, though. For learning the skeleton of the graph (UGE), the improvement obtained on the real cytoflow data is more substantial than on the synthetic data; see the left panel of Figure 6.16. This is a consequence of the fact that on the synthetic data, Bayesian networks show already a strong performance on learning the skeleton of the network, leaving not much room for further improvement. On the cytoflow data, on the other hand, the performance is much poorer. Consequently, the integration of prior knowledge leads to a more substantial improvement. When taking the edge directions into consideration

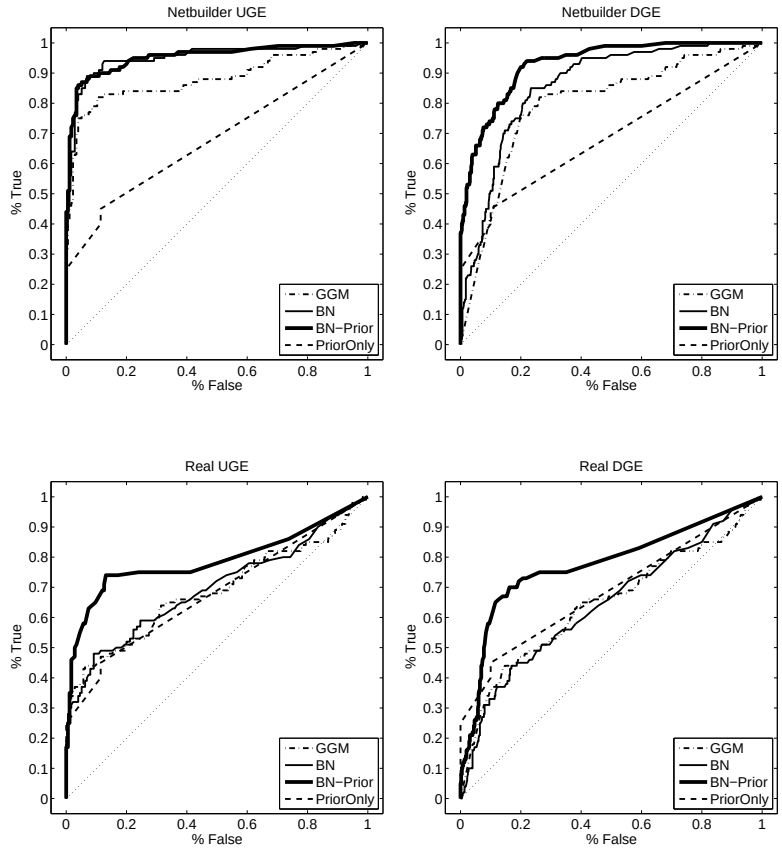


Figure 6.16: **Reconstruction of the Raf signalling pathway.** The figure evaluates the accuracy of inferring the Raf signalling network from cytometry data (bottom row) and from simulated Netbuilder data (top row), each combined with prior information from KEGG. This evaluation was carried out twice: with and without taking the edge direction into account (UGE: undirected graph evaluation, left column; DGE: directed graph evaluation, right column). Four machine learning methods were compared: Bayesian Networks without prior knowledge (BNs), Graphical Gaussian Models without prior knowledge (GGMs), Bayesian Networks with prior knowledge from KEGG (BNPrior), and prior knowledge from KEGG only (PriorOnly). In the latter case, the elements of the prior knowledge matrix (introduced in Section 6.2.1) were computed from equation (6.41). The ROC curves presented are the mean ROC curves obtained by averaging the results over five different data sets. The resulting areas under the ROC curves are as follows. Simulated data: DGE: GGM=0.795, BN=0.852 , BNPrior=0.929, PriorOnly=0.685; UGE: GGM=0.879, BN=0.952, BNPrior=0.948, PriorOnly=0.679; Flow cytometry data: DGE: GGM=0.645, BN=0.644, BNPrior=0.744, PriorOnly=0.685; UGE: GGM=0.686, BN=0.697, BNPrior=0.791, PriorOnly=0.679;

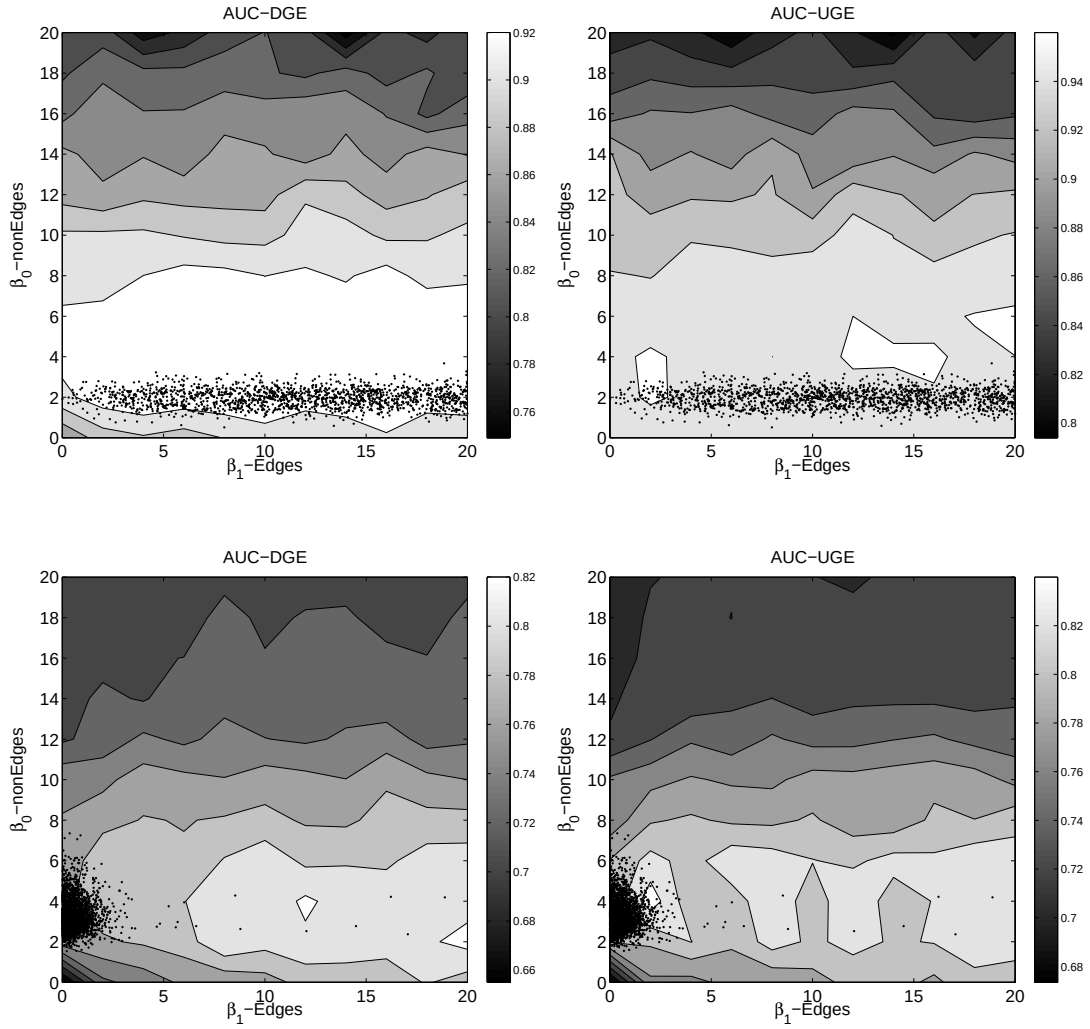


Figure 6.17: **Learning the hyperparameters associated with the prior knowledge from KEGG on simulated Netbuilder data and real flow cytometry data.** The grey shading of the contour plots represents the mean area under the ROC curve (AUC value) – averaged over five different data sets – as a function of the fixed values of the hyperparameters β_0 and β_1. The black dots show the values of these hyperparameters that were sampled in the MCMC simulations. The top row shows the results obtained on the simulated data. The bottom row shows the results obtained on the real flow cytometry protein concentrations. The left column shows the results for the directed graph evaluation (DGE), while the column on the right shows the results obtained when ignoring edge directions and only taking the skeleton of the network into account (UGE: undirected graph evaluation).

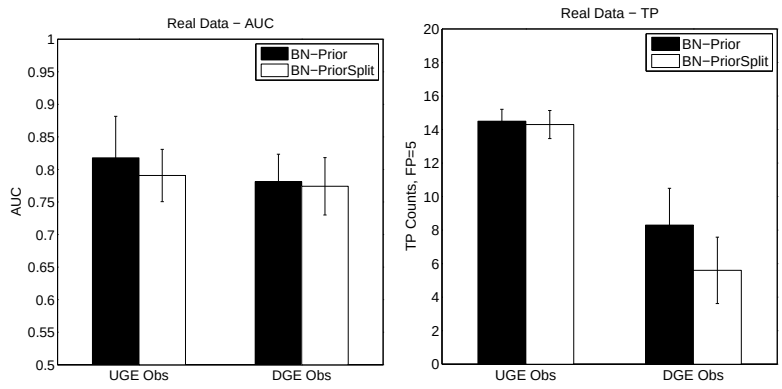


Figure 6.18: **Comparison between methods.** Here we compare the two different ways of incorporating one source of prior biological knowledge with gene expression data. The black bars show the case where there is only one hyperparameter which accounts for both presence and absence of edges. The white bars show the case where there are two hyperparameters one associated with the presence of edges and the other associated with the absence of edges. The results are obtained applying the methods to the real data. The left panel shows the AUC scores and the right panel shows the TP counts for both DGE and UGE scoring metrics.

(DGE), the proposed Bayesian integration scheme outperforms all other methods on the synthetic data; see Figure 6.16, top right. This result is consistent with what has been discussed in the Introduction section: when learning Bayesian networks from non-dynamical non-interventional data (as considered here) without prior knowledge, there is inherent uncertainty about the direction of edges owing to intrinsic symmetries within network equivalence classes; see Section 3.2.3. These symmetries are broken by the inclusion of prior knowledge; hence the improvement in the prediction performance. This improvement is also observed on the real cytoflow data (Figure 6.16, bottom right), but to a lesser extent. Although the area under the ROC curve related to the Bayesian integration scheme exceeds that of all other ROC curves, the prediction based on prior knowledge alone shows a steeper slope in the very left region of the false-positive axis. This implies that for very high values of the threshold on the edge scores, a network learned from prior knowledge alone is more accurate than a network learned with any of the three methods that make use of the data. While the resulting network itself would not be particularly interesting – it would only contain a very small

number (3 or 4) of the highest scoring edges – this observation is interesting nevertheless, and can be explained as follows. The discrepancy between the UGE and DGE scores indicates that the Bayesian network learns the skeleton of the graph more accurately than the direction of the interactions, with some of the edge directions systematically inverted. A possible explanation are errors in the gold standard network as discussed in Section 6.5.2.4. Such as yet unaccounted feedback loops could explain systematic deviations between the predicted and the gold standard network, not only because the structure of a Bayesian network is constrained to be acyclic, but also because we ultimately don't have a reliable gold standard to assess the quality of the predictions. This example points to a fundamental problem inherent in any evaluation based solely on real biological data, and illustrates clearly the advantage of our combined evaluation based on both laboratory and simulated data.

It is obviously of interest to test how well the inference of the hyperparameters β_0 and β_1 works, especially as this inference depends on the partition function $Z(\beta_0, \beta_1)$ of Equation (6.45), which can only be computed approximately; see Equation (6.6). To this end, we repeated the MCMC simulations for a large set of fixed values of β_0 and β_1, selected from the grid $[0, 20] \times [0, 20]$. For each pair of fixed values (β_0, β_1), we sampled BNs from the posterior distribution with MCMC, and evaluated the network reconstruction accuracy using the evaluation criteria described in Section 6.5.2.3. We compared these results with the proposed Bayesian inference scheme, where both hyperparameters and networks are simultaneously sampled from the posterior distribution with the MCMC scheme discussed in Section 6.2.2. The results are shown in Figure 6.17. The grey shading of the contour plots indicates the network reconstruction accuracy in terms of the directed (DGE: left panels) and undirected (UGE: right panels) graph evaluation, obtained from the synthetic (top panels) and real cytometry data (bottom panels). The black dots show the hyperparameter values sampled with the MCMC

simulations. While the distribution of β_0, the hyperparameter associated with the non-edges, is fairly peaked, the distribution of β_1, the hyperparameter associated with the edges, is rather diffuse. This diffusion is particularly noticeable on the synthetic data. However, even on the real cytometry data, the distribution of β_1 has a long tail, with values being sampled across the whole permissible spectrum. An inspection of the prior knowledge matrix B extracted from KEGG according to Equation (6.41) reveals that the prior knowledge associated with the energy function E_1 – Equation (6.43) – accounts for only 25% of the true edges in the gold standard network of Figure 4.1, while the prior knowledge associated with the energy function E_0 – Equation (6.42) – accounts for 92% of the non-edges. Consequently, it appears that E_0 captures more relevant information for network reconstruction than E_1, which is reflected by the tighter distribution of the respective hyperparameter. The location of the sampled values of the hyperparameters β_0 and β_1 falls into the region of high network reconstruction scores. This suggests that the proposed Bayesian sampling scheme succeeds in finding hyperparameter values that lead to good network reconstructions. A certain deviation from the optimal reconstruction would be expected owing to the approximation made for computing the partition function; see Equation (6.50). However, his deviation is small for both scores (UGE and DGE) on the synthetic data, and for the UGE score on the cytometry data. A noticeable deviation occurs for the DGE score on the cytometry data, though; see Figure 6.17, bottom left panel. This deviation indicates a systematic mismatch between the DGE score and the posterior probability of the hyperparameters, which suggests that the cytometry data do not support all the edge directions in the gold standard network of Figure 4.1. Two possible explanations are either wrong edge directions in the gold standard network, or the existence of as yet unaccounted feedback loops, in confirmation of what has been discussed above.

Another interesting comparison is between the two different ways of incorpo-

rating biological prior knowledge with gene expression data. Previously, in Section 6.2, we presented the method where each source of biological prior knowledge has one associated hyperparameter that accounts for both presence and absence of the edges. In the present section we introduced a modification to the way the source of prior biological knowledge is incorporated into the inference. Here each source of prior biological knowledge has two hyperparameters, one associated with the presence of edges and the other associated with the absence of edges. Figure 6.18 presents the results of the two methods applied to the data from flow cytometry experiments. This data set is described in Section 4.2. The results are all similar apart from the TP counts when considering the edge directions (DGE). The DGE TP-score is smaller for the case where the presence and absence of edges are considered separately. This difference does not appear for the UGE score indicating that the skeleton of the network is learnt but some edges directions are wrong. This difference also does not appear when looking into the AUC scores indicating that the wrong directed edges are present for very low values of FPs.

These results suggest that more flexibility in the presentation of the prior knowledge does not automatically guarantee a performance improvement. One of the reasons for this lack of improvement is presumably related to the fact that most of the useful prior information was contained in the absence of edges, whereas only little information was contained in the presence of interactions (as suggested by Figure 6.17). The decision of whether an edge is present or absent depends on the choice of the threshold, though, which was rather arbitrarily set to a fixed value of 0.5; see equations (6.48) and (6.49). A different choice of the threshold parameter might have led to a smaller disparity between the two subsets of edges with respect to the information content, which suggests that sampling this parameter from the posterior distribution with MCMC might have led to a clearer performance enhancement. This further suggests, on a more general

basis, that the flexibility and presentation of the prior knowledge about network structures, e.g. related to the subdivision of nodes and edges into subgroups, could be included in the MCMC scheme, which would provide an interesting avenue for future research.

6.7 Discussion

The work presented here is based on pioneering work by Imoto et al. (2003a) on learning gene regulatory networks from expression data and biological prior knowledge, which has recently found a variety of applications (Tamada et al., 2003; Nariai et al., 2004; Tamada et al., 2005; Imoto et al., 2006). The idea is to express the available prior knowledge in terms of an energy function, from which a prior distribution over network structures is obtained in the form of a Gibbs distribution. The hyperparameter of this distribution, which corresponds to an inverse temperature in statistical physics, represents the weight associated with the prior knowledge relative to the data. Our work complements the work of Imoto et al. (2003a) in various respects. We have extended the framework to multiple sources of prior knowledge; we have derived and tested an MCMC scheme for sampling networks and hyperparameters simultaneously from the posterior distribution; we have elucidated intrinsic features of this scheme from an idealized network population amenable to a closed-form derivation of the posterior distribution; and we have assessed the viability of the proposed Bayesian inference approach on various synthetic and real-world data.

Our findings can be summarized as follows. When including two sources of prior knowledge of similar nature, the marginal posterior distributions of the associated hyperparameters are similar (Figure 6.9). When the two sources of prior knowledge are different, higher weight is assigned to the prior that is more consistent with the data (Figure 6.11). When including an irrelevant prior with

vacuous information, its influence will be automatically suppressed (Figure 6.12) in that the marginal posterior distribution of the corresponding hyperparameter is shifted towards zero. The irrelevant prior is not completely switched off, though. This would correspond to a delta distribution sitting at zero, which is never observed, not even for the worst-case scenario of prior knowledge that is in complete contradiction to the true network and the data (Figure 6.8c). To elucidate this unexpected behaviour, we carried out two types of analysis. In the first case, we considered an idealized population of network structures for which the prior distribution could be computed in closed form (Equation 6.37). In the second case, we considered networks composed of a small number of nodes (Figure 6.4), for which the partition function of Equation 6.10, and hence the prior distribution over networks structures (Equation 6.9), could be numerically computed by exhaustive enumeration of all possible structures. Both types of analysis reveal that the posterior distribution over hyperparameters contains a flat plateau (Figure 6.8a-b), which accounts for our seemingly counter-intuitive observations.

We evaluated the accuracy of reconstructing the Raf protein signalling network, which has been extensively studied in the literature. To this end, we combined protein concentrations from cytometry experiments with prior knowledge from the KEGG pathway database. The findings of our study clearly demonstrate that the proposed Bayesian inference scheme outperforms various alternative methods that either take only the cytometry data or only the prior knowledge from KEGG into account (Figure 6.13). We inspected the values of the sampled hyperparameters. Encouragingly, we found that their range was close to the optimal value that maximizes the network reconstruction accuracy (Figure 6.14). A small systematic deviation would be expected owing to the approximation we have made for computing the partition function of the prior (Equations 6.6 and 6.16). Interestingly, a comparison between real and simulated cytometry data –

Figure 6.14 versus Figure 6.15 – revealed that the small bias only occurred in the former case. This suggests that other confounding factors, like errors in the gold-standard network and as yet unaccounted feedback loops, might have a stronger effect than the approximation made for computing the partition function.

A certain shortcoming of the proposed method is the intrinsic asymmetry between *prior knowledge* and *data*, which manifests itself in the fact that the hyperparameters of the prior are inferred from the data. Ultimately, the prior knowledge is obtained from some data also; for instance, prior knowledge about TF binding sites is obtained from immunoprecipitation data. A challenging topic for future research, hence, is to treat both *prior* and *data* on a more equal footing, and to develop more systematic methods of postgenomic data integration.

Chapter 7

Integrating data sets

This is joint work with Dirk Husmeier, submitted as part of an invited paper to be considered for possible publication in a special issue of the Journal of Bioinformatics and Computational Biology.

7.1 Introduction

The assumption so far has been that the molecular biological system of interest can be characterized by a unique regulatory network. What we are actually aiming to infer, though, are the active parts of this network, which may differ under different experimental conditions. To illustrate this point, consider a transcription factor that potentially upregulates a group of genes further downstream in the regulatory chain. If the experimental conditions are chosen such that the gene coding for this transcription factor is never expressed itself, then the respective subnetwork will never be activated, and hence cannot be inferred from the data. When aiming to infer regulatory networks related to an organism's immune system, we would expect certain pathways to be activated only upon infection, and remaining invisible when gene expression profiles are only taken in the healthy state. In fact, the analysis in Chapter 2 related to the challenging of macrophages with interferon gamma (IFNγ) and viral infection has revealed differences in the

active pathways under the conditions of viral infection, IFNγ treatment, and viral infection plus IFNγ treatment. This suggests that a regulatory network is not an immutable entity, but may vary in response to changes in the experimental and/or environmental conditions.

When aiming to reconstruct a network from gene expression profiles taken under different experimental conditions, there seem to be two principled approaches we may pursue. The first is to ignore the changes in the experimental conditions altogether and merge the data into one monolithic set. The problem with this approach is that it inevitably blurs the differences between the different conditions and thereby obscures the biological insight we are aiming to gain; for instance, we would not be able to tell the difference between the state of a network in infected, healthy, and IFNγ-treated cells. The second approach is to keep the data obtained under different conditions separate, and to infer separate regulatory networks active under these different conditions. While this approach has the potential to reveal the differences between the regulatory networks in different states, e.g. infection versus treatment, it will almost inevitably result in a considerable reduction in statistical power and reconstruction accuracy. Current postgenomic data sets are usually sparse, e.g. the number of microarray experiments biologists can afford to carry out is usually limited to the order of a few dozens. As discussed in Chapter 6, this limitation compromises the extent to which networks can be reconstructed. Breaking a sparse data set up into smaller units will inevitably aggravate this situation, and increase the uncertainty about inferred network structures.

In the present work we aim to pursue a compromise between the two extreme procedures described above. The motivation is given by the insight gained from Chapter 2. Although we found differences between the active pathways under the different conditions of infection and treatment with IFNγ, the networks shared considerable features they had in common. Our conjecture is that this holds

in general, and that a cell's regulatory networks, while potentially transitioning between different active states in response to different external cues, share substantial features owing to a common generic network architecture. Our objective is to formulate this proposition mathematically so as to integrate it into the probabilistic modelling process.

As it turns out, this objective can be achieved by a modification of the probabilistic model described in Chapter 6. Recall that the objective of Chapter 6 was the integration of explicit prior knowledge into the inference scheme by softly constraining the inferred network to be similar to the a-priori known network. In modification of this scheme we now propose to learn separate regulatory networks from disjunct gene expression data, but tying these networks together by softly constraining them to be similar to a shared underlying generic network. This approach overcomes the rigidity of the first scenario described above, which would obscure the differences between the network states in different experimental conditions. By sharing information between the different network states, the problem of the second scenario described above should be averted, that is, the statistical power and accuracy of the reconstruction should be considerably enhanced.

7.2 Methodology

In order to integrate information from I different data sets $(\mathcal{D}_1 \dots \mathcal{D}_I)$ obtained under different experimental conditions we use the probabilistic graphical model presented in Figure 7.1. Each data set $(\mathcal{D}_1 \dots \mathcal{D}_I)$ is associated with its own hyperparameter $(\beta_1, \dots, \beta_I)$ and network structure $(\mathcal{M}_1, \dots, \mathcal{M}_I)$. The latent graph $\mathcal{M}^\star$, which is not directly associated with the data, leads to a coupling between the individual network structures $(\mathcal{M}_1, \dots, \mathcal{M}_I)$ and encourages them to be similar. Note that Figure 7.1 constitutes a hierarchical Bayesian model, in which the β_is and $\mathcal{M}^\star$ correspond to hyperparameters that determine the prior

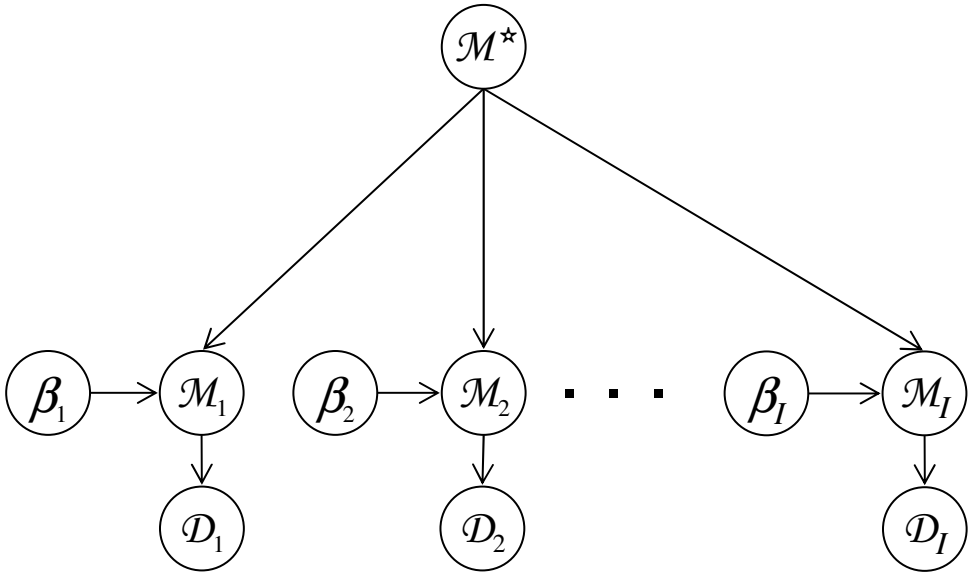


Figure 7.1: **Probabilistic model for learning active subnetworks under different experimental conditions.** $(\mathcal{D}_1 \ldots \mathcal{D}_I)$ are data sets obtained under different experimental conditions. Each of these data sets is associated with its own hyperparameter $(\beta_1, \ldots, \beta_I)$ and network structure $(\mathcal{M}_1, \ldots, \mathcal{M}_I)$. The hypernetwork $\mathcal{M}^\star$ leads to a coupling between the individual network structures $(\mathcal{M}_1, \ldots, \mathcal{M}_I)$ and encourages them to be similar.

distribution on the network structures $\mathcal{M}_i$s. Further note that $\mathcal{M}^\star$ is not just a variable, but a complex entity representing a whole network itself. We therefore refer to $\mathcal{M}^\star$ as the hypernetwork.

The joint probability of the probabilistic graphical model of Figure 7.1 is given by:

$$P(\mathcal{M}_1, \ldots, \mathcal{M}_I, \mathcal{D}_1 \ldots \mathcal{D}_I, \beta_1, \ldots, \beta_I, \mathcal{M}^\star) = \prod_{i=1}^{I} P(\mathcal{D}_i|\mathcal{M}_i) P(\mathcal{M}_i|\beta_i, \mathcal{M}^\star) P(\beta_i) P(\mathcal{M}^\star) \tag{7.1}$$

where the prior distribution over network structures, $P(\mathcal{M}_i|\beta_i, \mathcal{M}^\star)$, takes the form of a Gibbs distribution:

$$P(\mathcal{M}_i|\beta_i, \mathcal{M}^\star) = \frac{e^{-\beta_i(|\mathcal{M}_i - \mathcal{M}^\star|)}}{Z(\beta_i, \mathcal{M}^\star)}. \tag{7.2}$$

Recall that the hyperparameter β_i corresponds to an inverse temperature in statistical physics, and the term $|\mathcal{M}_i - \mathcal{M}^\star|$ measures the similarity between the

graphs $\mathcal{M}_i$ and $\mathcal{M}^\star$; see Equation (6.1) in Section 6.2. This introduces a coupling between the individual networks $\mathcal{M}_i$: deviations between $\mathcal{M}_i$ and $\mathcal{M}^\star$ are penalized, which implies an indirect penalty for deviations between $\mathcal{M}_i$ and $\mathcal{M}_k$, $i \neq k$. The denominator in (7.2) is a normalizing constant, also known as the partition function:

$$Z(\beta_i, \mathcal{M}^\star) = \sum_{\mathcal{M}_i \in \mathbb{M}} e^{-\beta_i(|\mathcal{M}_i - \mathcal{M}^\star|)} \tag{7.3}$$

where $\mathbb{M}$ is the set of all valid network structures. The summation over all possible models $\mathcal{M}_i$ can be performed efficiently using Equation (6.6), as discussed in the text below that equation.

The hyperparameter β_i can be interpreted as a factor that indicates the strength of the influence of the hypernetwork $\mathcal{M}^\star$ relative to the data. For $\beta_i \to 0$, the prior distribution defined in Equation (7.2) becomes flat and uninformative about the network structure. Conversely, for $\beta_i \to \infty$, the prior distribution becomes sharply peaked, forcing the network structure $\mathcal{M}_i$ to be similar to the hypernetwork $\mathcal{M}^\star$.

7.3 MCMC sampling scheme

Our goal is to sample all network structures $\mathcal{M}_i$, all the hyperparameters β_i and the hypernetwork $\mathcal{M}^\star$ from the posterior distribution. In order to achieve this objective we propose new structures $\mathcal{M}_{i_{\text{new}}}$ from the proposal distribution $Q_i(\mathcal{M}_{i_{\text{new}}}|\mathcal{M}_{i_{\text{old}}})$, new hyperparameters from the proposal distribution $R_i(\beta_{i_{\text{new}}}|\beta_{i_{\text{old}}})$ and a new hypernetwork from the proposal distribution $W(\mathcal{M}^\star_{\text{new}}|\mathcal{M}^\star_{\text{old}})$. We then accept these moves according to the standard Metropolis-Hastings update rule (Hastings, 1970) with the following acceptance

probability:

$$A = \min \left\{ \prod_{i=1}^{I} \frac{P(\mathcal{D}_i, \mathcal{M}_{i_{\text{new}}}, \beta_{i_{\text{new}}}, \mathcal{M}^{\star}_{\text{new}}) Q_i(\mathcal{M}_{i_{\text{old}}} | \mathcal{M}_{i_{\text{new}}}) R_i(\beta_{i_{\text{old}}} | \beta_{i_{\text{new}}})}{P(\mathcal{D}_i, \mathcal{M}_{i_{\text{old}}}, \beta_{i_{\text{old}}}, \mathcal{M}^{\star}_{\text{old}}) Q_i(\mathcal{M}_{i_{\text{new}}} | \mathcal{M}_{i_{\text{old}}}) R_i(\beta_{i_{\text{new}}} | \beta_{i_{\text{old}}})} \times \frac{W(\mathcal{M}^{\star}_{\text{old}} | \mathcal{M}^{\star}_{\text{new}}) P(\beta_{i_{\text{new}}}) P(\mathcal{M}^{\star}_{\text{new}})}{W(\mathcal{M}^{\star}_{\text{new}} | \mathcal{M}^{\star}_{\text{old}}) P(\beta_{i_{\text{old}}}) P(\mathcal{M}^{\star}_{\text{old}})}, 1 \right\} \tag{7.4}$$

For symmetric proposal distributions $R_i(\beta_{i_{\text{new}}}|\beta_{i_{\text{old}}})$ and $W(\mathcal{M}^{\star}_{\text{new}}|\mathcal{M}^{\star}_{\text{old}})$ this expression simplifies to:

$$A = \prod_{i=1}^{I} \frac{P(\mathcal{D}_i, \mathcal{M}_{i_{\text{new}}}, \beta_{i_{\text{new}}}, \mathcal{M}^{\star}_{\text{new}}) Q_i(\mathcal{M}_{i_{\text{old}}} | \mathcal{M}_{i_{\text{new}}}) P(\beta_{i_{\text{new}}}) P(\mathcal{M}^{\star}_{\text{new}})}{P(\mathcal{D}_i, \mathcal{M}_{i_{\text{old}}}, \beta_{i_{\text{old}}}, \mathcal{M}^{\star}_{\text{old}}) Q_i(\mathcal{M}_{i_{\text{new}}} | \mathcal{M}_{i_{\text{old}}}) P(\beta_{i_{\text{old}}}) P(\mathcal{M}^{\star}_{\text{old}})} \tag{7.5}$$

The prior distribution $P(\mathcal{M}^{\star})$ can be chosen in a manner that explicit biological prior knowledge is included as discussed in Chapter 6. However, for the sake of simplicity of the notation and in order to focus on the coupling aspects of the proposed method, we assume that both prior distributions $P(\beta_i)$ and $P(\mathcal{M}^{\star})$ are uniform; this leads to the following simplification of the expression:

$$A = \prod_{i=1}^{I} \frac{P(\mathcal{D}_i | \mathcal{M}_{i_{\text{new}}}) P(\mathcal{M}_{i_{\text{new}}} | \beta_{i_{\text{new}}}, \mathcal{M}^{\star}_{\text{new}}) Q_i(\mathcal{M}_{i_{\text{old}}} | \mathcal{M}_{i_{\text{new}}})}{P(\mathcal{D}_i | \mathcal{M}_{i_{\text{old}}}) P(\mathcal{M}_{i_{\text{old}}} | \beta_{i_{\text{old}}}, \mathcal{M}^{\star}_{\text{old}}) Q_i(\mathcal{M}_{i_{\text{new}}} | \mathcal{M}_{i_{\text{old}}})} \tag{7.6}$$

where we have expanded the joint probability according to the conditional independence relations shown in Figure 7.1. Note that the $\mathcal{M}_i$s, as opposed to $\mathcal{M}^{\star}$, need to be proper DAGs. For this reason, we include the corresponding Hastings factor – the last term in the equation – as it is not necessarily equal to one. In our simulations, to be discussed below, we have used edge-based proposal moves: the creation, deletion and reversal of an edge. When enforcing these moves to be valid, that is, to lead to proper DAGs, the two proposal probabilities do not necessarily cancel out and have therefore to be explicitly computed; see Section 3.2.2.2 for further details.

In order to increase the acceptance probability and, hence, mixing and convergence of the Markov chain, we break the move up into submoves. First we propose new structures for each of the networks $\mathcal{M}_i$ in turn, while keeping all the other variables fixed. The new structures are accepted with the following

acceptance probabilities:

$$\begin{aligned} A(\mathcal{M}_{i_{\text{new}}}|\mathcal{M}_{i_{\text{old}}}) = & \min\left\{\frac{P(\mathcal{D}_i|\mathcal{M}_{i_{\text{new}}})P(\mathcal{M}_{i_{\text{new}}}|\beta_i,\mathcal{M}^\star)Q_i(\mathcal{M}_{i_{\text{old}}}|\mathcal{M}_{i_{\text{new}}})}{P(\mathcal{D}_i|\mathcal{M}_{i_{\text{old}}})P(\mathcal{M}_{i_{\text{old}}}|\beta_i,\mathcal{M}^\star)Q_i(\mathcal{M}_{i_{\text{new}}}|\mathcal{M}_{i_{\text{old}}})},1\right\} \\ = & \min\left\{\frac{P(\mathcal{D}_i|\mathcal{M}_{i_{\text{new}}})e^{-\beta_i(|\mathcal{M}_{i_{\text{new}}}-\mathcal{M}^\star|)}Q_i(\mathcal{M}_{i_{\text{old}}}|\mathcal{M}_{i_{\text{new}}})}{P(\mathcal{D}_i|\mathcal{M}_{i_{\text{old}}})e^{-\beta_i(|\mathcal{M}_{i_{\text{old}}}-\mathcal{M}^\star|)}Q_i(\mathcal{M}_{i_{\text{new}}}|\mathcal{M}_{i_{\text{old}}})},1\right\} \end{aligned} \tag{7.7}$$

where (7.2) has been used. Next we propose new values for the trade-off hyperparameters β_i. Each of the trade-off hyperparameters is accepted with the following acceptance probability:

$$\begin{aligned} A(\beta_{i_{\text{new}}}|\beta_{i_{\text{old}}}) = & \min\left\{\frac{P(\mathcal{M}_i|\beta_{i_{\text{new}}},\mathcal{M}^\star)}{P(\mathcal{M}_i|\beta_{i_{\text{old}}},\mathcal{M}^\star)},1\right\} \\ = & \min\left\{\frac{e^{-\beta_{i_{\text{new}}}(|\mathcal{M}_i-\mathcal{M}^\star|)}Z(\beta_{i_{\text{old}}},\mathcal{M}^\star)}{e^{-\beta_{i_{\text{old}}}(|\mathcal{M}_i-\mathcal{M}^\star|)}Z(\beta_{i_{\text{new}}},\mathcal{M}^\star)},1\right\}. \end{aligned} \tag{7.8}$$

Finally a new hypernetwork is proposed and accepted with acceptance probability:

$$\begin{aligned} A(\mathcal{M}^\star_{\text{new}}|\mathcal{M}^\star_{\text{old}}) = & \min\left\{\prod_{i=1}^{I}\frac{P(\mathcal{M}_i|\beta_i,\mathcal{M}^\star_{\text{new}})}{P(\mathcal{M}_i|\beta_i,\mathcal{M}^\star_{\text{old}})},1\right\} \\ = & \min\left\{\prod_{i=1}^{I}\frac{e^{-\beta_i(|\mathcal{M}_i-\mathcal{M}^\star_{\text{new}}|)}Z(\beta_i,\mathcal{M}^\star_{\text{old}})}{e^{-\beta_i(|\mathcal{M}_i-\mathcal{M}^\star_{\text{old}}|)}Z(\beta_i,\mathcal{M}^\star_{\text{new}})},1\right\}. \end{aligned} \tag{7.9}$$

To illustrate the plausibility of this sampling scheme, consider the sampling of the hyperparameters β_i according to equation (7.8). We would assume that for a network $\mathcal{M}_i$ that consistently differs from the hypernetwork $\mathcal{M}^\star$, the corresponding hyperparameter β_i should be driven to small values (indicating weak coupling), while, conversely, β_i should be driven to large values (indicating strong coupling) when a network $\mathcal{M}_i$ is consistently similar to $\mathcal{M}^\star$. This is indeed the case. In the first scenario, $|\mathcal{M}_i - \mathcal{M}^\star|$ tends to be large, and high values of β_i are repressed by the exponential term in (7.8). In the second scenario, $|\mathcal{M}_i - \mathcal{M}^\star|$ becomes small, and the exponential term tends towards a constant, indiscriminative with respect to selecting β_i. Note, however, that the partition function $Z(\beta_i, \mathcal{M}^\star)$ is a monotonically decreasing function in β_i, as seen from Figure 7.2. This monotonicity provides a penalty for small values of β_i, driving β_i up to high values, as expected.

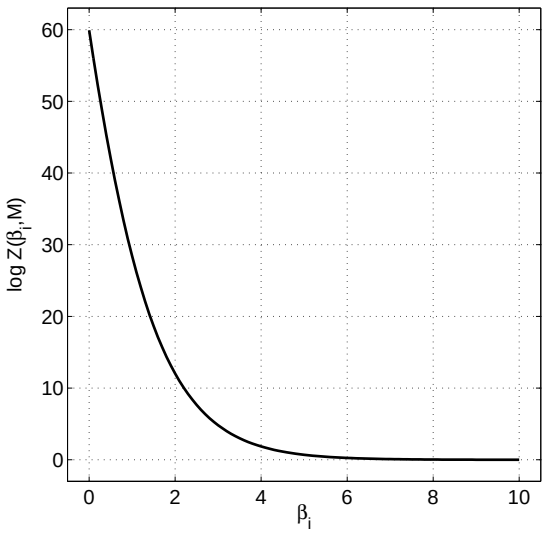


Figure 7.2: **Partition function example.** The figure shows a plot of the partition function $Z(\beta_i, \mathcal{M}^\star)$ as a function of the hyperparameter β_i for a fixed hypernetwork $\mathcal{M}^\star$, chosen to be the gold standard Raf network of Figure 4.1.

7.4 Data

We tested the proposed method on three types of data: linear Gaussian synthetic data, non-linear Netbuilder synthetic data, and real laboratory data from cytometry experiments. In each case, we coupled five individual data sets, corresponding to five experimental conditions. For the synthetic data, three of the data sets were generated from the gold-standard RAF regulatory network, shown in Figure 4.1. A fourth data set was generated from a slightly modified version of this network, in which the following four edges had been deleted: PKC → RAF, PKC → PKA, PKA → MEK, and PLCg → PIP2. An illustration of this network is shown in Figure 4.2. The deletion of these edges corresponds to changes in the active subpathways under different external conditions, as described above. As a fifth data set, we included a purely random data set. This corresponds to either a drastic change of the external conditions that deactivates the whole pathway, or to a flawed experiment that has corrupted the data. We want to investigate whether the proposed method succeeds in identifying this outlying data set and prevents it from adversely affecting the overall inference. We are also interested

in whether the proposed method can distinguish between the data from the gold-standard and the modified RAF regulatory network. The synthetic data sets we use in this chapter are explained in Chapter 4. Figure 4.3 shows a summary of all the data sets. Note that in this chapter we used only the observational data sets. For the cytometry data, we took four subsets of unintervened data, randomly selected from the data in Sachs et al. (2005) and pre-processed as described in Chapter 4. To these data we added a fifth data set, consisting of pure noise.

7.5 Results

7.5.1 Inferring the hyperparameters

Figure 7.3 shows various MCMC trace plots obtained on the linear Gaussian data, where the columns refer to different simulations. The first row shows trace plots of the log likelihood, while the remaining rows show trace plots of the hyperparameters β_i associated with the different data sets. The question of interest is whether the proposed method can identify the corrupted data set (pure noise), and distinguish between the data generated from the true network and those generated from the modified network. The first simulation (column 1) fails in this respect. In fact, the value of the hyperparameter β_{rand} associated with the corrupted data consistently exceeds the values of the other hyperparameters. However, the log likelihood is consistently low, suggesting that the MCMC simulations have not yet converged. This conjecture is corroborated by the second simulation, which shows a behaviour similar to the first simulation at the beginning, but then undergoes a sharp phase transition, during which β_{rand} is suddenly suppressed, while the other hyperparameters shoot up to high values. A concomitant transition in the log likelihood indicates that the Markov chain is escaping from a metastable low-probability state in which it was trapped. The two remaining simulations, corresponding to columns 3 and 4 of Figure 7.3, show a better convergence from

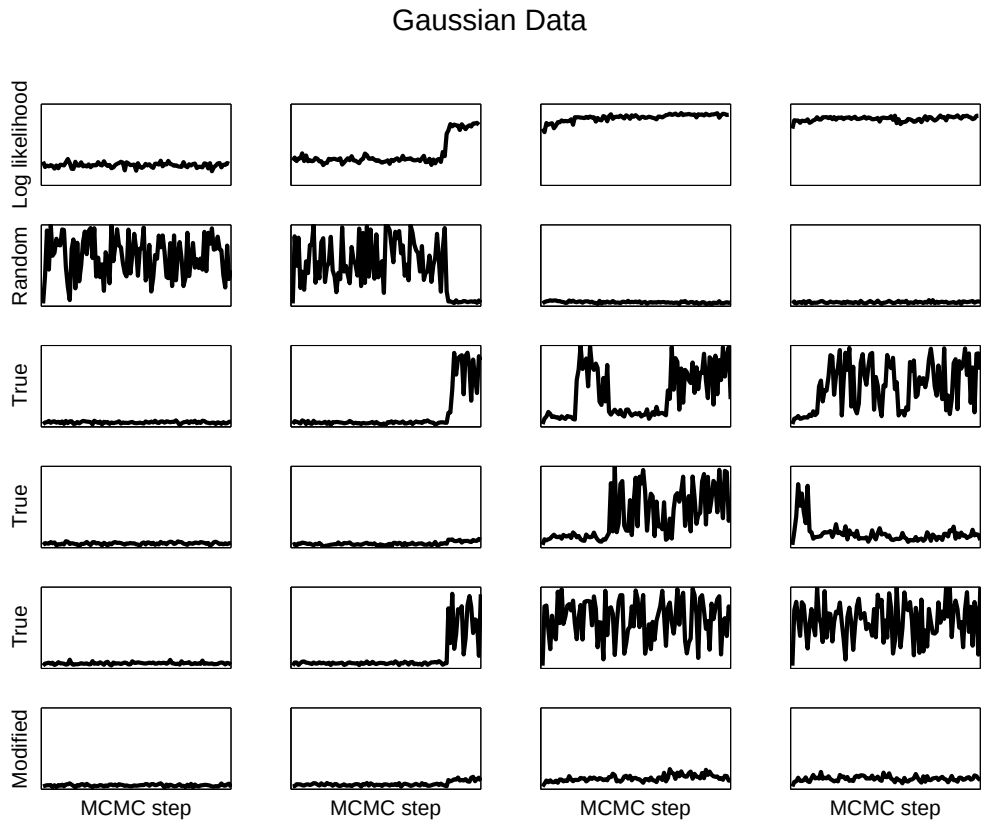


Figure 7.3: **MCMC trace plots for Gaussian data.** The columns represent different simulations. The first row shows trace plots of the log likelihood (computed from Equation (7.1)), while the remaining rows show trace plots of the hyperparameters β_i associated with the five different data sets used. These data sets are of different nature. $Random$: Corrupted data consisting of pure noise. $True$: Data sets generated from the gold-standard RAF network, shown in Figure 4.1. $Modified$: Data generated from the modified RAF network, in which four edges had been deleted; see Figure 4.2. The horizontal axes represent the number of MCMC interactions and have all the same scale. The number of MCMC steps for all simulations is 2×10^6 from which the first half is discarded as burn-in phase. The same scale was chosen for the vertical axes of the first row (log likelihoods). The vertical axes of the remaining rows (β_is) also have all the same scale which is the interval $[0, 30]$.

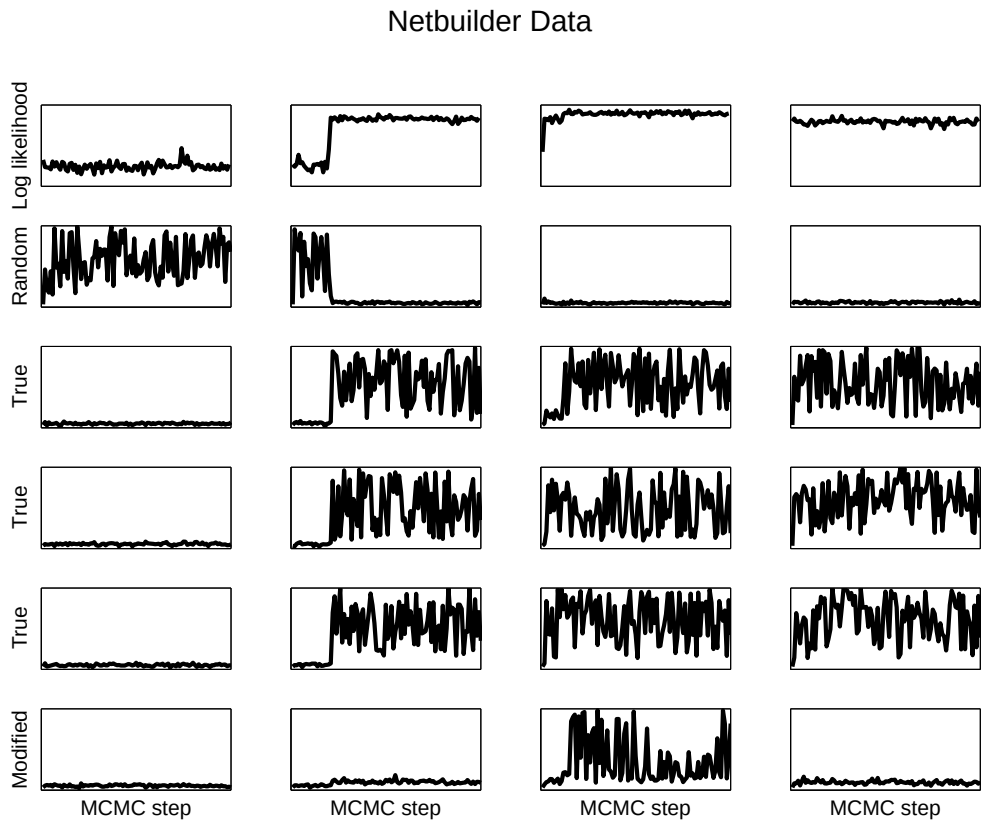
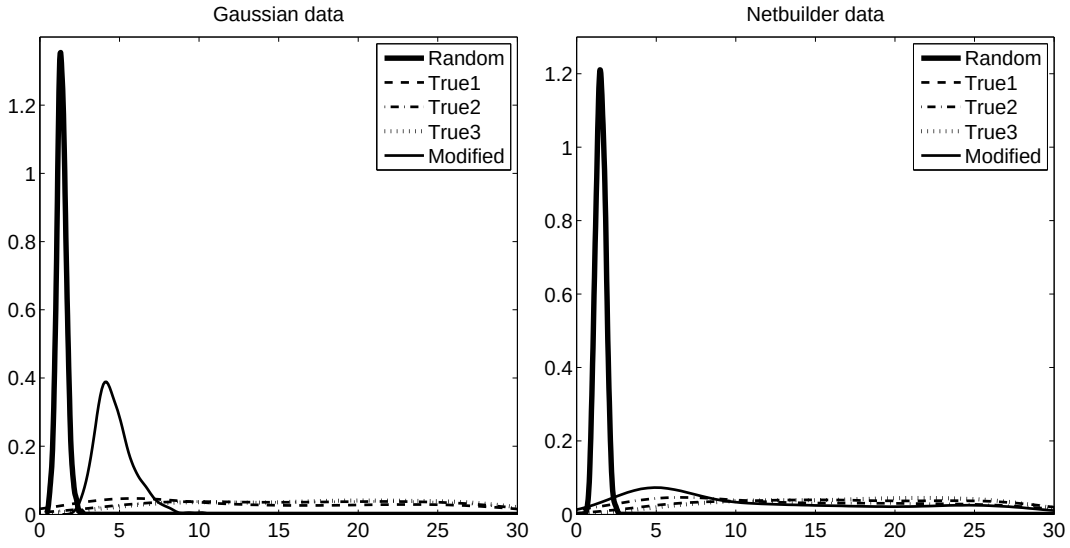


Figure 7.4: **MCMC trace plots for Netbuilder data.** The graphs correspond to those of Figure 7.3, but were obtained on the non-linear synthetic data rather than the Gaussian data. See the caption of Figure 7.3 for further details.



(a) (b)

Figure 7.5: **Posterior distributions of the hyperparameters.** These figures show the posterior distributions of the five hyperparameters $\beta_1, \ldots, \beta_5$, estimated with a kernel estimator applied to the samples obtained from the MCMC simulations with the best convergence characteristics. *Left panel:* linear Gaussian data. *Right panel:* non-linear data generated with Netbuilder.

the outset, with β_{rand} being consistently suppressed, and the hyperparameter associated with the modified network taking on values below those of the hyperparameters associated with the true network. A similar behaviour can be found in Figure 7.4, which was obtained from four MCMC simulations on the non-linear synthetic data. Figure 7.5 shows the estimated posterior distributions of the five hyperparameters for the best-converged MCMC simulations on both the linear and non-linear synthetic data. These plots suggest that the proposed method succeeds in identifying the corrupted data, whose associated hyperparameter is significantly suppressed, as well as the data generated from the modified network. In the latter case, the distribution of the respective hyperparameter is shifted to lower values than the distributions of the hyperparameters associated with the true network. The amount of shift varies between the two data sets, which we suspect is more related to different degrees of convergence of the Markov chains than intrinsic differences between the linear and non-linear data. The upshot of

this study is that the proposed method works successfully, but that convergence problems of the MCMC simulations can become an issue. One problem in our first set of simulations was that we initialised all networks as empty graphs. This gives the hyperparameter associated with the corrupted data a certain 'head-start': high-scoring networks inferred from the corrupted data will only contain a few edges, as there are no true associations between randomized nodes. This makes these networks similar to the hypernetwork (which was initialised as an empty graph), explaining the high value of β_{rand} at the beginning of some of our simulations. A better strategy is to pre-train the individual networks, e.g. using a greedy optimization, and to set the hypernetwork to their consensus network. This strategy was applied in the simulations of Figure 7.4, as opposed to Figure 7.3, and seems to have led to a modest improvement. There is, however, still substantial scope for the development of more efficient MCMC sampling schemes, as discussed below.

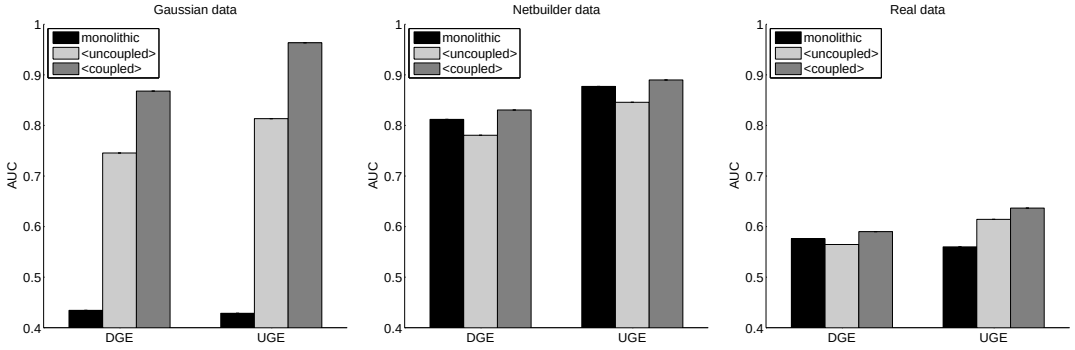
7.5.2 Network reconstruction

We are particularly interested in whether the proposed coupling scheme leads to any improvement in terms of network reconstruction accuracy over the two alternative approaches described above, namely: learning a single network from a merged, monolithic data set, and learning separate networks from the individual data sets without coupling. In what follows, we will refer to these methods as the *monolithic* and the *uncoupled* approach. To summarize the results succinctly, we take the area under the ROC curve as a performance criterion, both for the DGE and the UGE score, with larger areas indicating a better performance. The results are shown in Figure 7.6. The results presented for the uncoupled approach are the average AUC value of the 5 different simulations, one for each data set. They indicate that the proposed coupling scheme consistently outperforms the other two approaches. The improvement is most pronounced on the synthetic

	Gaussian		Netbuilder		Cytometry	
	DGE	UGE	DGE	UGE	DGE	UGE
Monolithic	0.43	0.42	0.82	0.88	0.57	0.56
Uncoupled	0.74	0.81	0.78	0.84	0.56	0.61
Coupled	0.86	0.96	0.83	0.89	0.59	0.64

Table 7.1: **Network reconstruction accuracy.** This table presents the AUC scores corresponding to the histograms in Figure 7.6. See the figure caption for further details.

Gaussian data. For these data, the control strength parameters associated with the edges in the regulatory network were different for each individual data set, which implies that even when the network structure itself did not change, the nature of the associated regulation processes could vary in both strength and sign (corresponding to an activation versus an inhibition). This explains the poor performance of the monolithic approach, which intrinsically does not allow for any such variation. The difference in performance is less pronounced for the non-linear synthetic data generated with Netbuilder, where only the instantiation of the noise rather than the parameters associated with the edges differed between different data sets. It appears that the slight performance improvement obtained with the proposed method is mainly a consequence of the inclusion of the corrupted data, whose influence gets suppressed as a consequence of the adaptation of the associated hyperparameter, as discussed above. For the cytometry data, the amount of performance improvement achieved with the proposed method lies between the two synthetic data sets, with the improvement being more noticeable for the reconstruction of the skeleton of the graph (UGE score) than the reconstruction of the edge directions (DGE score).

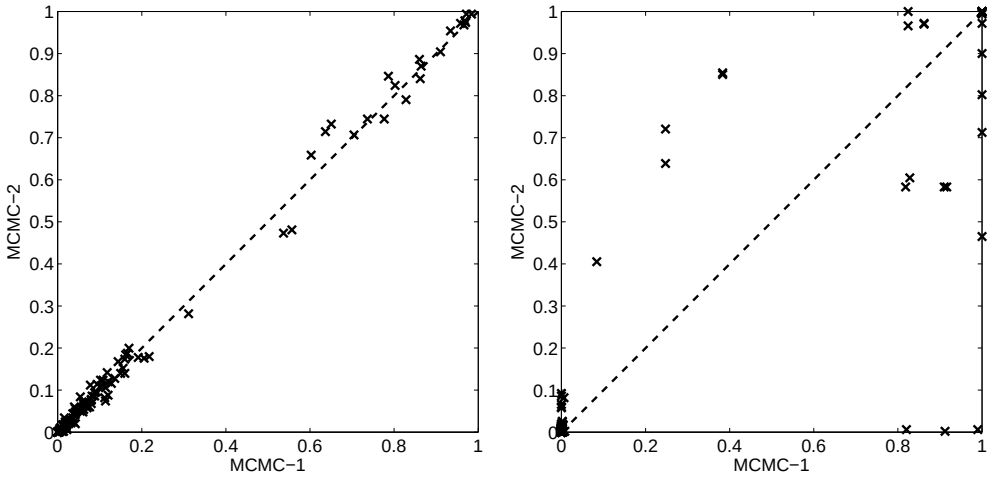


(a) (b) (c)

Figure 7.6: **Network reconstruction accuracy.** The histograms show a comparison of the network reconstruction accuracy in terms of AUC scores for three different methods: the monolithic approach (black), the uncoupled approach (light grey), and the proposed Bayesian coupling scheme (dark grey); see the main text for further details. The three panels correspond to different data sets: linear Gaussian synthetic data (left panel), non-linear synthetic data generated with Netbuilder (central panel), and protein concentrations from cytometry experiments (right panel). Each panel contains two histograms, evaluating only the reconstruction of the skeleton of the graph (UGE score) and additionally taking the edge direction into account (DGE score).

Network	Gaussian	Netbuilder	Cytometry
$\mathcal{M}_1$	54.4	54.4	55.9
$\mathcal{M}_2$	13.0	13.7	33.9
$\mathcal{M}_3$	13.4	15.3	24.9
$\mathcal{M}_4$	15.2	15.2	32.0
$\mathcal{M}_5$	12.8	13.5	31.6

Table 7.2: **MCMC acceptance ratios for uncoupled learning of network structures.** This table shows the MCMC acceptance ratios (in per cent) for the conventional scheme in which network structures $\mathcal{M}_1$ to $\mathcal{M}_5$ are learned independently from separate data sets. The higher acceptance ratio in the first row results from the fact that $\mathcal{M}_1$ was learned from random data, where the likelihood surface is flat. The higher acceptance ratio in the last column results from the smaller sample size of the cytometry data (20 rather than 100 exemplars), which again leads to a flatter likelihood surface.



(a) (b)

Figure 7.7: **MCMC convergence indication.** Each of the two panels shows a scatter plot of the marginal posterior probabilities of the edges, obtained from two separate MCMC simulations applied to a subset of the non-linear synthetic Netbuilder data. The left panel was obtained from the conventional approach, which aims to learn a separate Bayesian network from each subset of the data. The right panel was obtained from the proposed method, whereby Bayesian networks learned from different subsets of the data are coupled. The scatter plot was obtained from one of these coupled networks, corresponding to one of the $\mathcal{M}_i$'s in Figure 7.1. For the conventional scheme, shown in the left panel, there is a clear consistency between the results from the two independent MCMC simulations, that is, there is no indication of any convergence difficulties. For the proposed coupling scheme, however, the marginal posterior probabilities obtained from the two independent MCMC simulations differ, which indicates a lack of convergence.

Network	Gaussian	Netbuilder	Cytometry
$\mathcal{M}_1$	25.6	18.5	0.4
$\mathcal{M}_2$	1.2	2.3	14.2
$\mathcal{M}_3$	0.3	2.7	2.7
$\mathcal{M}_4$	0.05	2.4	13.3
$\mathcal{M}_5$	1.2	4.4	1.8
$\mathcal{M}^\star$	4.5	11.0	10^{-3}

Table 7.3: **MCMC acceptance ratios for the proposed Bayesian coupling scheme.** This table is to be compared with Table 7.2. It shows the MCMC acceptance ratios (in per cent) for learning five network structures $\mathcal{M}_1$ to $\mathcal{M}_5$ from five separate data sets. As opposed to Table 7.2, the networks are coupled via a hypernetwork $\mathcal{M}^\star$ according to the proposed coupling scheme illustrated in Figure 7.1. It is seen that as a consequence of this coupling, the MCMC acceptance ratios have substantially decreased.

7.5.3 Convergence of the Markov chains

A possible reason for the occasionally only modest performance improvement of the proposed method over the two alternative approaches is a lack of convergence of the MCMC simulations. Convergence problems have already been discussed in Subsection 7.5.1, and become more obvious in Figure 7.7. The panels in this figure show scatter plots of the marginal posterior probabilities of the edges obtained from two separate MCMC simulations, started from different initializations. The panel on the left was obtained from the conventional uncoupled MCMC scheme. The marginal posterior probabilities obtained from two independent simulations are very similar, indicating consistency of the predictions irrespective of the initialization. However, the panel on the right of Figure 7.7 – obtained from the proposed coupling scheme – shows a noticeable difference between the two independent MCMC simulations, which clearly indicates a lack of convergence. This behaviour was found consistently throughout our simulations. To shed more light onto the convergence characteristics, we computed the average acceptance ratios of the MCMC moves during the whole simulation. Table 7.2 shows the acceptance ratios for the conventional scheme without coupling. Table 7.3 shows the acceptance ratios for the proposed coupling scheme. A comparison between these two tables suggests that as a consequence of coupling, the acceptance ratios have significantly decreased. This can be understood intuitively in that as a result of coupling, a local modification of a network structure is not only penalised when moving into regions of lower posterior probability, but also when increasing the difference between the network structures. The result is a higher rigidity of the Markov chain, which shows poorer mixing and convergence than the uncoupled scheme. A possible approach to deal with this rigidity is to adopt a simulated annealing scheme. Alternatively, more sophisticated sampling schemes could be explored, as briefly discussed below.

7.6 Conclusion

We have proposed a Bayesian coupling scheme for learning gene regulatory networks from a combination of related data sets that were obtained under different experimental conditions and are therefore potentially associated with different active subpathways. The proposed coupling scheme is a compromise between two extreme scenarios: (1) learning networks from the different subsets separately, whereby no information between the different experiments is shared, and (2) learning networks from a monolithic fusion of the individual data sets, which does not provide any mechanism for uncovering differences between the network structures associated with the different experimental conditions. Our proposed method combines the flexibility of the first approach with the data merging aspect inherent in the second approach. The essential idea is that the networks associated with the different experimental conditions are softly constrained to be similar, where the strength of this constraint is defined by a hyperparameter that is automatically inferred from the data. Inference of these hyperparameters as well as the network structures is carried out in the Bayesian framework by approximately sampling from the posterior distribution with MCMC. We have tested the proposed method on three types of data related to the widely studied RAF signalling pathway: two synthetic data sets, generated from the gold-standard network either under a linear Gaussian distribution, or under a non-linear distribution using Netbuilder; and real protein concentrations from cytometry experiments. Our results can be summarized as follows. Given sufficient convergence of the MCMC simulations, a random data set deliberately included with the proper data is clearly detected. The hyperparameter associated with the random data is automatically set to very small values; this suggests that the proposed Bayesian coupling scheme is effective in switching off the influence of corrupted data. A data set generated from a modified network structure is also automatically detected. The associated hyperparameter is sampled from a distribution

placed between those associated with the random data and the data from the unmodified network, successfully distinguishing it from both. In terms of network reconstruction accuracy, the proposed Bayesian coupling scheme consistently outperformed the two competing approaches. The performance difference was most noticeable on those synthetic data where the individual data sets corresponded to different activation levels of the regulatory subpathways (owing to different settings of the interaction parameters). The difference was less pronounced when only adding corrupted data to data from unchanged experimental (cytometry data) or simulation (Netbuilder) conditions. A problem intrinsic to the proposed new scheme is a deterioration of the convergence and mixing of the Markov chain. In fact, some of the results presented here were obtained from MCMC simulations that had incompletely converged, suggesting that the performance improvement achieved with coupling could be further improved upon proper convergence. Unfortunately, this aspect has to be left to future research; owing to funding and visa restrictions, this thesis had to be completed within the prescribed period of studies. As future research, we will explore novel proposal moves, which swap substructures between the individual networks and the hypernetwork, allowing the latter to change in a more systematic way at (hopefully) a higher acceptance probability. The running of parallel Metropolis-coupled Markov chains, as described in Geyer (1991); Gilks et al. (1996) and successfully applied in phylogenetics Huelsenbeck and Ronquist (2001), will also be attempted, especially as it will allow the exploitation of modern PC clusters, and might offer ways to more efficiently design highly accepted proposal moves based on information obtained from the whole population of Markov chains (Laskey and Myers, 2003).

Chapter 8

Conclusion and future work

Since the DNA structure was unveiled in 1953 (Watson and Crick, 1953) and the central dogma of molecular biology was enunciated (Watson and Crick, 1958; Crick, 1970) there has been a rapid development in this field of research. Molecular biology has shifted to be a quantitative science where the understanding of complex molecular biological systems plays a key role. Among many studied problems in molecular biology one very attractive is the discovery of genetic regulatory networks from data. In this particular area the main aim is to discover how genes work together in an orchestrated way in order to keep living organisms healthy and alive. The knowledge about these intricate relationships between genes has the potential to unveil new solutions for various disease problems. The main focus of this thesis is to evaluate and improve mathematical and statistical methods that are used in the inference of genetic regulatory networks.

After an introduction in Chapter 1 we have moved to an example in Chapter 2 where the true network is assumed to be known and, thus, it can be used in conjunction with gene expression data in order to search for subpathways that are active under different experimental conditions. This example illustrates the usefulness of having the knowledge about network structures and how this knowledge can be used in conjunction with other data to improve our understanding about

molecular biological systems. In this example the resulting active subnetworks share many genes in common. One would have expected to see more discrete active networks for each experimental condition. One possible problem is that we were using a pathway composed of elements that were manually curated and, hence, is far from being complete. Thus, the differences in the active subnetworks are probably occurring in elements which our study does not cover. This emphasizes the need for better knowledge about network structures in order to improve their usefulness and reinforce the need for methods that can automatically learn network structures from data.

Having got a better idea about how important the discovery of biological network structures is we have advanced to Chapter 3 where the statistical theory of the methods that have been used in this thesis have been explored. Namely the methods we compared are: Relevance Networks, Graphical Gaussian Models and Bayesian Networks. The list of methods is far from being exhaustive but it covers popular methods of the machine learning community which are widely used. Moreover, a practical comparison amongst these methods is very useful since they were broadly explored theoretically and we can explore their advantages and drawbacks when used in practice. In Chapter 4 we have presented the data sets that have been used in the subsequent chapters and the evaluation criteria we have used to assess the performance of the algorithms.

After exploring the methods' theory we have moved to their practical comparison in Chapter 5. Theoretically the advantages and drawbacks of each method are well understood. What was still missing was a proper comparison of their practical use. One of the constraints when comparing these methods is the data. While simulated data is relatively easy to obtain, it is often very distinct from real measured data. There are many real data sets available but unfortunately most of these data sets have been measured from biological systems for which we do not have the complete knowledge about the network structure and, moreover,

they are generally very sparse. Hence, the comparison using real data is very difficult. A very nice exception to the stated problems with real data sets is the flow cytometry data generated in the experiment of Sachs et al. (2005). The network from which this data was measured was widely studied using traditional biological molecular methods and, therefore, the underlying accepted network is quite reliable. Furthermore, this data set has both observational and interventional measurements, which enriches our comparison. This real data set is also distinct from other available data sets as it measures protein concentrations whilst the majority of other available data sets are obtained from microarray experiments where mRNA concentrations are measured. In addition to the real data we generated simulated Gaussian and Netbuilder data in order to compare the performances. One shortcoming of this study is the fact that we only have data from one network structure. The ideal situation would have been to have data from a collection of networks structures, but real data sets with this quality are still very rare. In order to minimize this shortcoming we used a second network with a slightly modified structure for simulating data.

The main conclusion of the comparison study is that interventions are necessary in order to justify the use of BNs. In general if interventions are not available GGMs, which are much faster, gave similar results to BNs. BNs are superior to GGMs when considering interventions. They are more flexible and allow the proper inclusion of the information about the interventions. Interventions in BNs are mainly useful in order to resolve edge directions that cannot be learned due to the equivalence classes of BNs. Conversely, GGMs cannot make use of this extra information. Another important point to note is that in fact BNs can learn directed edges from passive observations without interventions only when v-structures are present. Therefore we expect that BNs would have a better performance over GGMs, even using only observational data, if the underlying structure contains more v-structures.

Unfortunately data sets with interventions are not available very often. Another way of exploring the full potential of BNs would be the use of extra information in order to resolve ambiguous edge directions from equivalence classes. This is the motivation for the study about the inclusion of biological prior knowledge in the reverse engineering of BNs.

In Chapter 6 we presented a methodology for including extra sources of information in the inference of genetic regulatory networks. As mentioned before expression data is often very sparse and noisy and the inclusion of extra knowledge is essential in order to increase the quality of the predicted networks. In our study we have improved and extended the method of Imoto et al. (2003a). We have treated data sets other than gene expression as biological prior knowledge and integrated them into the inference process through a prior distribution over network structures. The integration of extra knowledge is balanced through a trade-off hyperparameter that indicates how much of this information should be used. The trade-off hyperparameter is also learned from the data. In order to test the method we evaluated it on an idealized population of network structures for which the closed form expression of the relevant posterior distribution can be obtained. Moreover we have applied the method to real microarray time series data and to the flow cytometry data.

The results are very encouraging. The method is able to distinguish relevant biological prior knowledge from spurious information and we have shown that the value of the trade-off hyperparameter is close to the value of the optimal hyperparameter. We show that in the case where the biological prior knowledge is completely useless the hyperparameter is effectively switched off and this information does not influence the results. Furthermore, the method clearly outperformed all the other methods that we included in our comparison.

One shortcoming of the method is that it treats all the data other than gene expression data as biological prior knowledge. Ultimately one would like to in-

clude and treat all the data sets equally.

In Chapter 7 a method for the integration of data sets generated from the same biological system under different experimental conditions has been presented. The assumption is that a given underlying biological network has always the same topology, however, not all its components act in the same way under different experimental conditions. Thus, what can be inferred using the data collected from different experimental conditions are slightly different active subnetworks owing to the fact that some of their components may not be active or may be acting in a different way, e.g. activation vs inhibition.

There are two ways of using gene expression profiles generated from different experimental conditions in order to reconstruct networks. One way is to merge all the data in a monolithic data set and use it for inference. The other way is to use each of the data sets separately and infer different networks for each of these data sets. What we have proposed is a third way of integrating data sets from different experimental conditions. The rationale is that there is a generic network that shares various topological features with the different active networks under different experimental challenges. We have proposed learning separate regulatory networks from distinct gene expression data tying the learned networks to be similar to each other through an underlying generic network, which in our model corresponds to a hyperparameter in a hierarchical Bayesian model and is hence referred to as the "hypernetwork".

Although it is clear that the MCMC simulations with this method had not properly converged it was superior to the other options of using the data in all the simulations. The most contrasting result is the one from Gaussian generated data. The different data sets generated with the Gaussian distribution were generated from the same network topology but with different weights associated with their edges. The weights have different values (which represent strong vs. weak interactions) and different signs (which represent activations vs. inhibitions). Such a

contrast was observed to a lesser extent on the Netbuilder generated data and in the real data. The Netbuilder data was generated with all the weights associated with edges having the same value. All the real data sets effectively come from the same experimental condition and, therefore, it is expected that their parameters should be roughly the same.

While our method achieved a consistent improvement in terms of network reconstruction accuracy, there are certain problems with the MCMC convergence that need to be solved. One problem is that this method introduces new constraints on the networks to be sampled. These new constraints clearly slow down the mixing and convergence of the MCMC.

The discovery of whole regulatory networks or pathways from measured data is still in its infancy. Much progress has been made in the last few years but we are still far from the day where a whole biological network or pathway will be accurately discovered from data alone. There are many shortcomings that need to be addressed. The largest available type of gene profile measurements comes undoubtedly from microarray experiments. Although this technology is still being improved the data it generates is still generally very noisy and sparse. One important question is: Is it a reasonable assumption that the mRNA concentrations are proportional to protein activities? I believe that there is not a definitive answer to this question at the moment. Flow cytometry measurements solve the problem with the aforementioned assumption but they are very far away from reaching the coverage of genes that microarrays already have. Nowadays modern flow cytometers can measure the expression of 18 components at maximum. There is not a clear technical limit to the number of components that a flow cytometer can measure though.

On the modelling side we should not forget that a real biological system is a system for which our knowledge is still very limited. For instance there are still discussions about the role of the non-coding parts of DNA (Pearson, 2006;

Claverie, 2005). There is evidence to believe that the fine tuning of gene regulation is executed by the introns or micro RNAs that until recently were considered as "junk" DNA (Ying and Lin, 2004). This suggests that using only gene expression data in order to unveil genetic regulatory networks can be a very difficult task. Furthermore, even with our limited knowledge we know that the dynamic and complex real biological systems are far from being similar to the static and quite simple models that we use to represent them in our simulations. We are driven to use these somewhat more simple models due to computational restrictions and due to the fact that more refined models would need substantial improvements in data quality and quantity in order to become viable.

On the computing side there are many possibilities for future work. In all this thesis we were working with single processor computers and with relatively small networks. In order to work with larger networks it is definitely necessary the use of parallel computer clusters. A parallel computer cluster is a natural choice to deal with an intrinsically parallel processed system like a cell, where all the processes of its constituent components, e.g. related to transcription, translation etc., are inherently parallel. In fact, even with a rather small network, with only 11 nodes, we faced convergence problems in our simulations of Chapter 7 where the model is more complex and the need for parallelization of the simulations is evident. With parallel processors there are many possibilities that can be explored. For example, we foresee that one could use parallel processors to compute the score for each node of the network. Or another possibility would be to use parallel processors to run parallel Metropolis-coupled Markov chains or yet the combination of the two stated possibilities.

Having in mind all these problems it is clear that there is still much to be done in this area of reconstructing gene regulatory networks from postgenomic data.

Appendix A

Reversibility of MCMC moves

A.1 Moving Uniform with boundaries

Consider a parameter x' to be sampled in a MCMC move given that we have the actual value x. The proposal distribution is uniform over the interval $[x-l, x+l]$ with the constraint that $x' \in [0, \mathrm{MAX}]$. If the sampled value x' happens to be outside the allowed interval, the value is *reflected* back to the interval.

First lets check the case where the proposed x' could lie below the minimum value. A new value x^* is proposed from the interval $[x-l, x+l]$ and we can have two possible outcomes:

1. $x' = x^*$ if $x^* \geq 0$

2. $x' = -x^*$ if $x^* < 0$

If $x < l$ we need to consider the possibility of reflection. Note that the maximum value that can be reflected is $x - l$ which will be reflected as $l - x$. This gives us two regions on the interval $[x-l, x+l]$ with different probabilities. What separates these two regions is the point $l - x$. Values above $l - x$ are never reflected and values below $l - x$ can be reflected and are therefore twice as likely to occur. This happens because the values of the reflected region can be obtained

in two different ways: $x^* \in [0, l-x[$ or $x^* \in [x-l, 0]$. We have then the following probabilities for the interval $[x-l, x+l]$ according to the reflection.

- If $x' \leq l - x \Rightarrow P(x'|x) = \frac{1}{2l} + \frac{1}{2l} = \frac{1}{l}$
- If $x' > l - x \Rightarrow P(x'|x) = \frac{1}{2l}$.

We now need to compute the probabilities for the inverse move, $P(x|x')$:

- For the first part of the interval we have:

  $x' \leq l - x \Rightarrow P(x'|x) = \frac{1}{l}$

  Also:

  $x' \leq l - x \Rightarrow x \leq l - x' \Rightarrow P(x|x') = \frac{1}{l}$

  Hence:

  $P(x'|x) = P(x|x')$

- For the second part of the interval we have:

  $x' > l - x \Rightarrow P(x'|x) = \frac{1}{2l}$

  Also:

  $x' > l - x \Rightarrow x > l - x' \Rightarrow P(x|x') = \frac{1}{2l}$

  Hence:

  $P(x'|x) = P(x|x')$

With this we show that $P(x'|x) = P(x|x')$ and hence they cancel out in the Hastings ratio for the lower limit. Figure A.1 shows two examples, left panel shows the case where $x' \leq l - x$ and the right panel shows the case $x' > l - x$.

Now we investigate the case where the proposed x' can be above the maximum value, $x_{\max}$. Again we propose a new value x^* from the interval $[x-l, x+l]$ and we can have two cases:

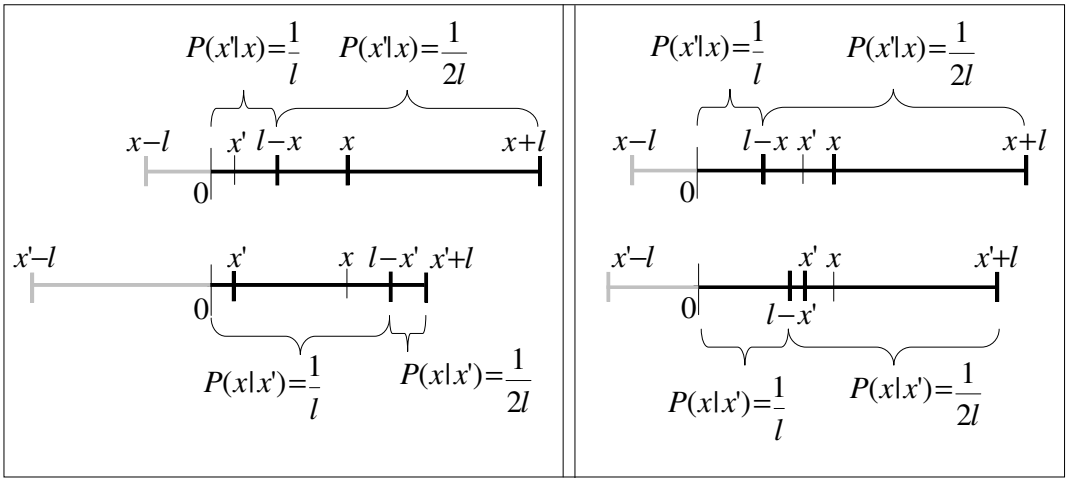


Figure A.1: **Moving Uniform lower limit.** Gray line represents the part of the interval which is outside the minimum limit. The upper part of both panels shows the original interval and the new sampled value x'. The bottom part of both panels shows the new interval set by the new value x' and where the original value x sits in this new interval. For both intervals the probabilities are defined and we can see from the graphs that in both cases $P(x'|x) = P(x|x')$. The left panel shows the case where $x' \leq l - x$ and the right panel shows the case where $x' > l - x$.

1. $x' = x^*$ if $x^* \leq x_{\max}$.

2. $x' = (2x_{\max} - x^*)$ if $x^* > x_{\max}$.

If $x + l > x_{\max}$ we need to consider the possibility of reflection in the upper limit. In this case the maximum value that can be reflected is $x + l$, which will be reflected as $2x_{\max} - (x + l)$. This is the point which separates the interval $[x - l, x + l]$ in two regions with a different probability for each. Values below $2x_{\max} - (x + l)$ are never reflected. Values above this point can be reflected and are twice as likely to happen. This happen because the values of the reflected region can be obtained in two different ways: $x^* \in]2x_{\max} - (x + l), x_{\max}]$ or $x^* \in [x_{\max}, x + l]$. We have therefore the following probabilities for the interval $[x - l, x + l]$ according to the reflection:

- If $x' \leq (2x_{\max} - (x + l)) \Rightarrow P(x'|x) = \frac{1}{2l}$
- If $x' > (2x_{\max} - (x + l)) \Rightarrow P(x'|x) = \frac{1}{2l} + \frac{1}{2l} = \frac{1}{l}$

We now need to compute the probabilities for the inverse move, $P(x|x')$:

- For the first part of the interval we have:

  $x' \leq (2x_{\max} - (x + l)) \Rightarrow P(x'|x) = \frac{1}{2l}$

  Also:

  $x' \leq (2x_{\max} - (x + l)) \Rightarrow x \leq 2x_{\max} - (x' + l) \Rightarrow P(x|x') = \frac{1}{2l}$

  Hence:

  $P(x'|x) = P(x|x')$

- For the second part of the interval we have:

  $x' > (2x_{\max} - (x + l)) \Rightarrow P(x'|x) = \frac{1}{l}$

  Also:

  $x' > (2x_{\max} - (x + l)) \Rightarrow x > (2x_{\max} - (x' + l)) \Rightarrow P(x|x') = \frac{1}{l}$

  Hence:

  $P(x'|x) = P(x|x')$

With this we show that $P(x'|x) = P(x|x')$ and hence they cancel out in the Hastings ratio for the upper limit. Figure A.2 shows two examples, left panel shows the case where $x' > (2x_{\max} - (x + l))$ and the right panel shows the case where $x' \leq (2x_{\max} - (x + l))$.

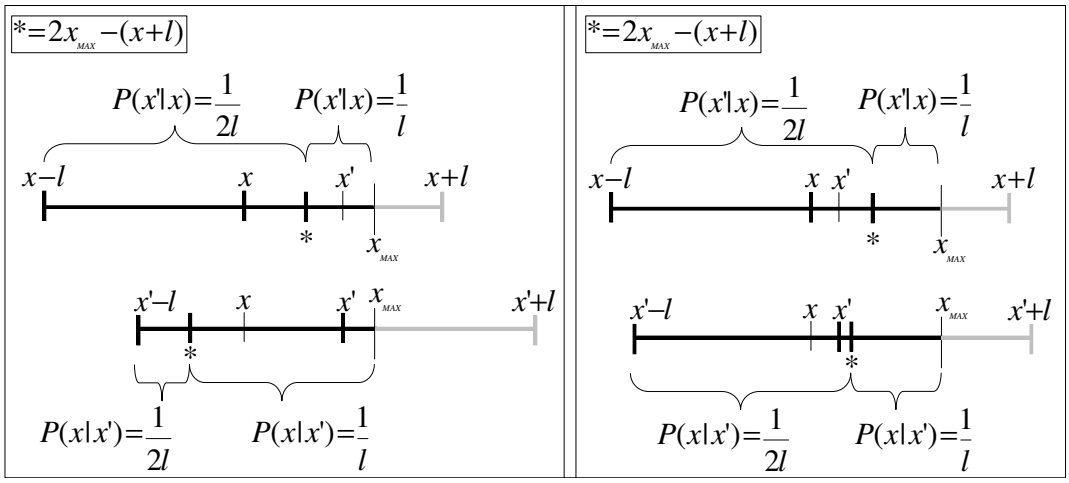


Figure A.2: **Moving Uniform upper limit.** Gray line represents the part of the interval which is outside the maximum limit. The upper part of both panels shows the original interval and the new sampled value x'. The bottom part of both panels shows the new interval set by the new value x' and where the original value x sits in this new interval. For both intervals the probabilities are defined and we can see from the graphs that in both cases $P(x'|x) = P(x|x')$. The left panel shows the case where $x' > (2x_{\text{max}} - (x + l))$ and the right panel shows the case where $x' \leq (2x_{\text{max}} - (x + l))$.

Appendix B

Comparisons' p-value tables

This appendix presents the results of a collaboration study with Marco Grzegorczyk and Dirk Husmeier, published as supplementary material in Werhli et al. (2006).

B.1 Cross-Method comparison of AUC scores

This section of the supplementary material provides nine tables, numbered from B.1 to B.9, which summarise and cross-compare the performances of the three Machine Learning methods under comparison in terms of the outputed *AUC* scores. Thereby for each table multiple rows indicate the four combinations of figure of merit (UGE and DGE) and data set type (observational and interventional). For each of these four combinations and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean μ[AUC] and the the standard deviations σ(AUC) of the five outputed *AUC* scores can be found. The last three columns provide one-sample t-test p-values p(.) for the hypothesis: H_0: $\mu[\text{AUC}(M_i)] = \mu[\text{AUC}(M_j)]$ against its two-sided alternative: H_1: $\mu[\text{AUC}(M_i)] \neq \mu[\text{AUC}(M_j)]$ given the combination indicated in the multiple row above. M_i and M_j represent the methods mentioned in the row and column. Low p-values p(.) indicate that there may be a significant difference in the AUC score between these two methods for the

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.8848	0.0543	-	0.8815	0.0079
GGM	0.8814	0.0373	0.8815	-	0.0015
RN	0.6809	0.0816	0.0079	0.0015	-
DGE - Observational					
BN	0.7817	0.0711	-	0.6704	0.0239
GGM	0.7967	0.0286	0.6704	-	0.0015
RN	0.6407	0.0635	0.0239	0.0015	-
UGE - Interventional					
BN	0.9661	0.0391	-	0.0024	0.0018
GGM	0.8203	0.0532	0.0024	-	0.0082
RN	0.7097	0.0541	0.0018	0.0082	-
DGE - Interventional					
BN	0.9796	0.0187	-	0.0002	0.0002
GGM	0.7488	0.0409	0.0002	-	0.0081
RN	0.6631	0.0421	0.0002	0.0081	-

Table B.1: **AUC score. Cross method comparison** Gaussian data sets. Original graph topology.

particular combination of figure of merit and data set type. In these cases it can be seen from the entries in the mean score column μ[AUC] which of the two methods performed (significantly) better than the other one.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9775	0.0345	-	0.0087	0.0013
GGM	0.8933	0.0583	0.0087	-	0.0043
RN	0.6987	0.0981	0.0013	0.0043	-
DGE - Observational					
BN	0.9487	0.0440	-	0.0012	0.0004
GGM	0.8257	0.0487	0.0012	-	0.0043
RN	0.6649	0.0814	0.0004	0.0043	-
UGE - Interventional					
BN	1.000	0.0000	-	0.0010	0.0014
GGM	0.8878	0.0293	0.0010	-	0.0199
RN	0.7436	0.0730	0.0014	0.0199	-
DGE - Interventional					
BN	0.9976	0.0038	-	0.0001	0.0004
GGM	0.8220	0.0001	0.0001	-	0.0196
RN	0.7021	0.0004	0.0004	0.0196	-

Table B.2: **AUC score. Cross method comparison** Gaussian data sets. V-structure graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.6904	0.0376	-	0.8754	0.2957
GGM	0.6854	0.0542	0.8754	-	0.6175
RN	0.6680	0.0546	0.2957	0.6175	-
DGE - Observational					
BN	0.6231	0.0564	-	0.5316	0.7276
GGM	0.6443	0.0419	0.5316	-	0.6139
RN	0.6307	0.0425	0.7276	0.6139	-
UGE - Interventional					
BN	0.7912	0.0335	-	0.0552	0.0003
GGM	0.7129	0.0559	0.0552	-	0.0010
RN	0.5686	0.0286	0.0003	0.0010	-
DGE - Interventional					
BN	0.6969	0.0676	-	0.4802	0.0076
GGM	0.6656	0.0437	0.4802	-	0.0010
RN	0.5533	0.0222	0.0076	0.0010	-

Table B.3: **AUC score. Cross method comparison** Real cytoflow data sets.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7901	0.0336	-	0.0764	0.0444
GGM	0.8143	0.0191	0.0764	-	0.0009
RN	0.7434	0.0081	0.0444	0.0009	-
DGE - Observational					
BN	0.6808	0.0703	-	0.0669	0.7977
GGM	0.7446	0.0150	0.0669	-	0.0010
RN	0.6893	0.0063	0.7977	0.0010	-
UGE - Interventional					
BN	0.7047	0.0221	-	0.0675	0.0076
GGM	0.7297	0.0183	0.0675	-	0.0410
RN	0.7537	0.0063	0.0076	0.0410	-
DGE - Interventional					
BN	0.8280	0.0097	-	0.0001	0.0000
GGM	0.6793	0.0144	0.0001	-	0.0468
RN	0.6973	0.0049	0.0000	0.0468	-

Table B.4: **AUC score. Cross method comparison** Nebuilder data sets low noise level($\sigma = 0.01$). Original graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9564	0.0273	-	0.0247	0.0469
GGM	0.8803	0.0656	0.0247	-	0.0909
RN	0.9323	0.0188	0.0469	0.0909	-
DGE - Observational					
BN	0.8572	0.0100	-	0.0288	0.0116
GGM	0.7957	0.0508	0.0288	-	0.0891
RN	0.8362	0.0146	0.0116	0.0891	-
UGE - Interventional					
BN	0.9346	0.0254	-	0.0188	0.0006
GGM	0.8300	0.0438	0.0188	-	0.1466
RN	0.8003	0.0082	0.0006	0.1466	-
DGE - Interventional					
BN	0.9678	0.0114	-	0.0004	0.0000
GGM	0.7574	0.0339	0.0004	-	0.1359
RN	0.7336	0.0064	0.0000	0.1359	-

Table B.5: **AUC score. Cross method comparison** Nebuilder data sets medium noise level($\sigma = 0.1$). Original graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9049	0.0150	-	0.2776	0.1310
GGM	0.8829	0.0486	0.2776	-	0.0750
RN	0.9163	0.0179	0.1310	0.0750	-
DGE - Observational					
BN	0.8208	0.0223	-	0.3024	0.8234
GGM	0.7979	0.0381	0.3024	-	0.0782
RN	0.8238	0.0139	0.8234	0.0782	-
UGE - Interventional					
BN	0.9053	0.0367	-	0.0168	0.0329
GGM	0.8571	0.0251	0.0168	-	0.7139
RN	0.8631	0.0273	0.0329	0.7139	-
DGE - Interventional					
BN	0.9219	0.0408	-	0.0013	0.0007
GGM	0.7776	0.0230	0.0013	-	0.7051
RN	0.7824	0.0212	0.0007	0.7051	-

Table B.6: **AUC score. Cross method comparison** Nebuilder data sets high noise level($\sigma = 0.3$). Original graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7845	0.0184	-	0.0055	0.0018
GGM	0.8529	0.0139	0.0055	-	0.0000
RN	0.7170	0.0094	0.0018	0.0000	-
DGE - Observational					
BN	0.7354	0.0467	-	0.0748	0.0558
GGM	0.7927	0.0117	0.0748	-	0.0000
RN	0.6801	0.0078	0.0558	0.0000	-
UGE - Interventional					
BN	0.7102	0.0156	-	0.0008	0.3208
GGM	0.7900	0.0180	0.0008	-	0.0110
RN	0.7280	0.0279	0.3208	0.0110	-
DGE - Interventional					
BN	0.8413	0.0052	-	0.0000	0.0001
GGM	0.7258	0.0143	0.0000	-	0.0115
RN	0.6773	0.0217	0.0001	0.0115	-

Table B.7: **AUC score. Cross method comparison** Nebuilder data sets low noise level($\sigma = 0.01$). V-structure graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9887	0.0114	-	0.0259	0.0002
GGM	0.9567	0.0294	0.0259	-	0.0024
RN	0.8513	0.0188	0.0002	0.0024	-
DGE - Observational					
BN	0.9674	0.0124	-	0.0002	0.0000
GGM	0.8788	0.0244	0.0002	-	0.0025
RN	0.7915	0.0156	0.0000	0.0025	-
UGE - Interventional					
BN	0.9927	0.0085	-	0.0019	0.0000
GGM	0.8277	0.0565	0.0019	-	0.0395
RN	0.7483	0.0257	0.0000	0.0395	-
DGE - Interventional					
BN	0.9944	0.0040	-	0.0002	0.0000
GGM	0.7547	0.0436	0.0002	-	0.0390
RN	0.6931	0.0200	0.0000	0.0390	-

Table B.8: **AUC score. Cross method comparison** Nebuilder data sets medium noise level($\sigma = 0.1$). V-structure graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9332	0.0454	-	0.0437	0.0020
GGM	0.9038	0.0562	0.0437	-	0.2289
RN	0.9154	0.0460	0.0020	0.2289	-
DGE - Observational					
BN	0.8745	0.0452	-	0.0888	0.0931
GGM	0.8350	0.0466	0.0888	-	0.2135
RN	0.8447	0.0381	0.0931	0.2135	-
UGE - Interventional					
BN	0.9788	0.0090	-	0.0163	0.0018
GGM	0.8677	0.0630	0.0163	-	0.2972
RN	0.8214	0.0474	0.0018	0.2972	-
DGE - Interventional					
BN	0.9393	0.0406	-	0.0047	0.0013
GGM	0.7861	0.0489	0.0047	-	0.2943
RN	0.7500	0.0368	0.0013	0.2943	-

Table B.9: **AUC score. Cross method comparison** Nebuilder data sets high noise level($\sigma = 0.3$). V-structure graph topology.

B.2 Cross-Method comparison of True Positive counts

This section of the supplementary material provides nine tables, numbered from B.10 to B.18, which summarise and cross-compare the performances of the three Machine Learning methods under comparison in terms of the true positive counts TP obtained when accepting 5 false positive counts (FP=5). Thereby in analogy to the last section for each table multiple rows indicate the four combinations of figure of merit (UGE and DGE) and data set type (observational and interventional). For each of these four combinations and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean μ[TP] and the the standard deviations σ(TP) of the five true positive counts TP obtained for 5 false positive counts can be found in the first columns. The last three columns provide one-sample t-test p-values p(.) for the hypothesis: H_0: $\mu[\mathrm{TP}(M_i)] = \mu[\mathrm{TP}(M_j)]$ against its two-sided alternative: H_1: $\mu[\mathrm{TP}(M_i)] \neq \mu[\mathrm{TP}(M_j)]$ given the combination indicated in the multiple row above. M_i and M_j represent the methods mentioned in the row and column. Low p-values p(.) indicate that there may be a significant difference in the TP counts between these two methods for the particular combination of figure of merit and data set type. In these cases it can be seen from the entries in the mean score column μ[TP] which of the two methods performed (significantly) better than the other one. In contrast to the *AUC* score cross-method comparison this alternative true positive count cross-method comparison concentrates on a fixed inverse specificity point of the ROC curve.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.8	2.1	-	0.1662	0.0010
GGM	14.8	2.7	0.1662	-	0.0012
RN	8.1	2.5	0.0010	0.0012	-
DGE - Observational					
BN	4.9	1.5	-	0.6885	0.0042
GGM	4.7	1.1	0.6885	-	0.0705
RN	3.8	1.3	0.0042	0.0705	-
UGE - Interventional					
BN	18.5	2.4	-	0.0074	0.0028
GGM	13.2	2.0	0.0074	-	0.0011
RN	6.5	2.7	0.0028	0.0011	-
DGE - Interventional					
BN	18.4	2.6	-	0.0005	0.0005
GGM	5.2	0.7	0.0005	-	0.0036
RN	1.8	1.3	0.0005	0.0036	-

Table B.10: **TP counts score. Cross method comparison** Gaussian data sets. Original graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.6	0.5	-	0.0270	0.0000
GGM	11.8	2.7	0.0270	-	0.0024
RN	5.8	1.4	0.0000	0.0024	-
DGE - Observational					
BN	11.3	1.2	-	0.0000	0.0001
GGM	3.8	1.0	0.0000	-	0.1951
RN	3.0	0.6	0.0001	0.1951	-
UGE - Interventional					
BN	16.0	0.0	-	0.0025	0.0001
GGM	12.9	1.0	0.0025	-	0.0054
RN	7.1	1.3	0.0001	0.0054	-
DGE - Interventional					
BN	15.8	0.4	-	0.0000	0.0000
GGM	5.5	0.0	0.0000	-	0.0008
RN	3.7	0.4	0.0000	0.0008	-

Table B.11: **TP counts score. Cross method comparison** Gaussian data sets. V-structure graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	9.5	2.0	-	0.7489	0.7174
GGM	9.6	1.6	0.7489	-	0.3046
RN	9.3	1.6	0.7174	0.3046	-
DGE - Observational					
BN	3.3	2.3	-	0.1369	0.1369
GGM	5.1	0.9	0.1369	-	NaN
RN	5.1	0.9	0.1369	NaN	-
UGE - Interventional					
BN	11.1	1.3	-	0.0951	0.0099
GGM	9.6	1.1	0.0951	-	0.0204
RN	7.1	1.1	0.0099	0.0204	-
DGE - Interventional					
BN	6.9	1.1	-	0.0065	0.0009
GGM	4.1	1.1	0.0065	-	0.0093
RN	1.7	0.4	0.0009	0.0093	-

Table B.12: **TP counts score. Cross method comparison** Real cytoflow data sets.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	11.0	2.0	-	0.2577	0.0366
GGM	12.0	1.2	0.2577	-	0.0040
RN	6.9	1.4	0.0366	0.0040	-
DGE - Observational					
BN	2.8	1.3	-	0.0077	0.0890
GGM	5.1	0.7	0.0077	-	0.0016
RN	0.8	0.8	0.0890	0.0016	-
UGE - Interventional					
BN	7.9	0.7	-	0.0008	0.0000
GGM	5.2	0.3	0.0008	-	0.0000
RN	2.0	0.0	0.0000	0.0000	-
DGE - Interventional					
BN	8.4	1.2	-	0.0019	0.0001
GGM	3.7	0.4	0.0019	-	0.0001
RN	0.0	0.0	0.0001	0.0001	-

Table B.13: **TP counts score. Cross method comparison.** Nebuilder data sets low noise level($\sigma = 0.01$). Original graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	18.1	1.1	-	0.0161	0.1216
GGM	16.5	1.1	0.0161	-	0.5291
RN	16.8	1.6	0.1216	0.5291	-
DGE - Observational					
BN	7.2	1.5	-	0.0673	0.0673
GGM	5.5	0.0	0.0673	-	NaN
RN	5.5	0.0	0.0673	NaN	-
UGE - Interventional					
BN	17.7	0.7	-	0.0046	0.0003
GGM	13.6	1.5	0.0046	-	0.0002
RN	8.0	1.7	0.0003	0.0002	-
DGE - Interventional					
BN	17.3	0.7	-	0.0000	0.0000
GGM	5.4	0.2	0.0000	-	0.0000
RN	1.2	0.7	0.0000	0.0000	-

Table B.14: **TP counts score. Cross method comparison.** Nebuilder data sets medium noise level($\sigma = 0.1$). Original graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.5	1.7	-	0.4468	0.2756
GGM	14.8	2.9	0.4468	-	0.0213
RN	16.6	2.3	0.2756	0.0213	-
DGE - Observational					
BN	4.1	2.0	-	0.5158	0.3844
GGM	4.7	1.1	0.5158	-	0.3739
RN	5.1	0.9	0.3844	0.3739	-
UGE - Interventional					
BN	16.0	1.6	-	0.0890	0.0143
GGM	14.5	1.5	0.0890	-	0.3672
RN	13.6	1.5	0.0143	0.3672	-
DGE - Interventional					
BN	14.1	4.5	-	0.0052	0.0073
GGM	5.5	0.0	0.0052	-	0.3739
RN	5.0	1.1	0.0073	0.3739	-

Table B.15: **TP counts score. Cross method comparison.** Nebuilder data sets high noise level($\sigma = 0.3$). Original graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	9.8	0.8	-	1.0000	0.0007
GGM	9.8	0.8	1.0000	-	0.0001
RN	5.2	0.3	0.0007	0.0001	-
DGE - Observational					
BN	3.9	0.7	-	0.3419	0.0019
GGM	4.5	0.9	0.3419	-	0.0020
RN	1.5	0.0	0.0019	0.0020	-
UGE - Interventional					
BN	7.2	0.4	-	0.4263	0.0102
GGM	7.5	0.4	0.4263	-	0.0078
RN	5.1	0.9	0.0102	0.0078	-
DGE - Interventional					
BN	6.6	0.4	-	0.0001	0.0000
GGM	3.4	0.2	0.0001	-	0.0002
RN	2.0	0.0	0.0000	0.0002	-

Table B.16: **TP counts score. Cross method comparison.** Nebuilder data sets low noise level($\sigma = 0.01$). V-structure graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.6	0.4	-	0.0876	0.0000
GGM	14.7	1.0	0.0876	-	0.0001
RN	9.3	0.4	0.0000	0.0001	-
DGE - Observational					
BN	12.0	1.5	-	0.0006	0.0007
GGM	5.5	0.0	0.0006	-	0.1079
RN	4.8	0.8	0.0007	0.1079	-
UGE - Interventional					
BN	15.7	0.4	-	0.0002	0.0005
GGM	12.5	0.5	0.0002	-	0.0021
RN	8.9	1.0	0.0005	0.0021	-
DGE - Interventional					
BN	15.4	0.7	-	0.0000	0.0000
GGM	4.9	0.8	0.0000	-	0.1302
RN	3.8	1.0	0.0000	0.1302	-

Table B.17: **TP counts score. Cross method comparison.** Nebuilder data sets medium noise level($\sigma = 0.1$). V-structure graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	14.2	1.1	-	0.0474	0.1087
GGM	13.2	1.0	0.0474	-	0.3375
RN	13.6	0.8	0.1087	0.3375	-
DGE - Observational					
BN	7.7	2.0	-	0.0714	0.0440
GGM	5.5	0.0	0.0714	-	0.2663
RN	5.0	0.9	0.0440	0.2663	-
UGE - Interventional					
BN	14.9	0.2	-	0.0093	0.0277
GGM	12.5	1.1	0.0093	-	0.5913
RN	12.8	1.4	0.0277	0.5913	-
DGE - Interventional					
BN	12.3	1.7	-	0.0009	0.0009
GGM	5.5	0	0.0009	-	NA
RN	5.5	0	0.0009	NA	-

Table B.18: **TP counts score. Cross method comparison.** Nebuilder data sets high noise level($\sigma = 0.3$). V-structure graph topology.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.8848	0.9661	0.0264
GGM	0.8814	0.8203	0.0683
RN	0.6809	0.7097	0.5285
DGE			
BN	0.7817	0.9796	0.0003
GGM	0.7967	0.7488	0.0643
RN	0.6407	0.6631	0.5285

Table B.19: **AUC score. Observational versus Interventional data.** Gaussian data sets. Original graph topology.

B.3 Comparison: observational vs. interventional data - AUC

This section of the supplementary material provides nine tables, numbered from B.19 to B.27, which compare the performance of each Machine Learning method on pure observational data with its performance on interventional data sets in terms of the outputed *AUC* scores. Thereby for each of the nine tables multiple rows indicate the two different figures of merit (UGE and DGE). For each figure of merit and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean of the five AUC scores on pure observational data μ[AUC|OBS] as well as the mean of the five AUC scores on interventional data μ[AUC|INT] are given in the first two columns. Subsequently the hypothesis H_0: μ[AUC|OBS] = μ[AUC|INT] was tested against its two-sided alternative H_1: μ[AUC|OBS] $\neq$ μ[AUC|INT] using two-sample t-tests. The p-values p(.) of these tests can be found in the last column. Low p-values p(.) indicate that there may be a significant difference in the mean AUC score obtained for pure observational and interventional data sets for the method mentioned in the row and the particular figure of merit. In these cases it can be seen from the entries in the mean score columns μ[AUC|OBS] and μ[AUC|INT] on which data set type the method performed (significantly) better.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.9775	1.0000	0.1822
GGM	0.8933	0.8878	0.8565
RN	0.6987	0.7436	0.4355
DGE			
BN	0.9487	0.9976	0.0384
GGM	0.8257	0.8220	0.8820
RN	0.6649	0.7021	0.4355

Table B.20: **AUC score. Observational versus Interventional data.** Gaussian data sets. V-structure graph topology.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.6904	0.7912	0.0021
GGM	0.6854	0.7129	0.4534
RN	0.6680	0.5686	0.0069
DGE			
BN	0.6231	0.6969	0.0974
GGM	0.6443	0.6656	0.4532
RN	0.6307	0.5533	0.0069

Table B.21: **AUC score. Observational versus Interventional data.** Real cytoflow data sets.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.7901	0.7047	0.0014
GGM	0.8143	0.7297	0.0001
RN	0.7434	0.7537	0.0553
DGE			
BN	0.6808	0.8280	0.0017
GGM	0.7446	0.6793	0.0001
RN	0.6893	0.6973	0.0553

Table B.22: **AUC score. Observational versus Interventional data.** Netbuilder data sets low noise level ($\sigma = 0.01$). Original graph topology.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.9464	0.9346	0.4974
GGM	0.8803	0.8300	0.1918
RN	0.9323	0.8003	0.0000
DGE			
BN	0.8572	0.9678	0.0000
GGM	0.7957	0.7574	0.1979
RN	0.8362	0.7336	0.0000

Table B.23: **AUC score. Observational versus Interventional data.** Netbuilder data sets medium noise level ($\sigma = 0.1$). Original graph topology.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.9049	0.9053	0.9813
GGM	0.8829	0.8571	0.3242
RN	0.9163	0.8631	0.0066
DGE			
BN	0.8208	0.9219	0.0013
GGM	0.7979	0.7776	0.3228
RN	0.8238	0.7824	0.0066

Table B.24: **AUC score. Observational versus Interventional data.** Netbuilder data sets high noise level ($\sigma = 0.3$). Original graph topology.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.7845	0.7102	0.0001
GGM	0.8529	0.7900	0.0003
RN	0.7170	0.7280	0.4271
DGE			
BN	0.7354	0.8413	0.0010
GGM	0.7927	0.7258	0.0000
RN	0.6801	0.6773	0.7986

Table B.25: **AUC score. Observational versus Interventional data.** Netbuilder data sets low noise level ($\sigma = 0.01$). V-structure graph topology.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.9887	0.9927	0.5464
GGM	0.9567	0.8277	0.0019
RN	0.8513	0.7483	0.0001
DGE			
BN	0.9674	0.9944	0.0017
GGM	0.8788	0.7547.	0.0005
RN	0.7915	0.6931	0.0000

Table B.26: **AUC score. Observational versus Interventional data.** Netbuilder data sets medium noise level ($\sigma = 0.1$). V-structure graph topology.

Method	μ[AUC|OBS]	μ[AUC|INT]	p(.)
UGE			
BN	0.9332	0.9788	0.0583
GGM	0.9038	0.8677	0.3669
RN	0.9154	0.8214	0.0129
DGE			
BN	0.8745	0.9393	0.0443
GGM	0.8350	0.7861	0.1442
RN	0.8447	0.7500	0.0040

Table B.27: **AUC score. Observational versus Interventional data.** Netbuilder data sets high noise level ($\sigma = 0.3$). V-structure graph topology.

B.4 Comparison: observational vs. interventional data - TP

This section of the supplementary material provides nine tables, numbered from B.28 to B.36, which compare the performance of each Machine Learning method on pure observational data with its performance on interventional data sets in terms of the obtained true positive counts (TP) when accepting five false negative counts (FP=5). As in the last section for each of the nine tables multiple rows indicate the two different figures of merit (UGE and DGE). For each figure of merit and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean of the five TP counts on pure observational data μ[TP|OBS] as well as the mean of the five TP counts on interventional data μ[TP|INT] are given in the first two columns. Subsequently the hypothesis H_0: μ[TP|OBS] = μ[TP|INT] was tested against its two-sided alternative H_1: μ[TP|OBS] $\neq$ μ[TP|INT] using two-sample t-tests. The p-values p(.) of these tests can be found in the last column. Low p-values p(.) indicate that there may be a significant difference in the mean TP count outputed for pure observational and interventional data sets for the method mentioned in the row and the particular figure of merit. In these cases it can be seen from the entries in the mean TP count columns μ[TP|OBS] and μ[TP|INT] on which data set type the method performed (significantly) better.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	15.8	18.5	0.0971
GGM	14.8	13.2	0.3152
RN	8.1	6.5	0.3553
DGE			
BN	4.9	18.4	0.0000
GGM	4.7	5.2	0.4094
RN	3.8	1.8	0.0386

Table B.28: **TP counts score. Observational versus Interventional data.** Gaussian data sets. Original graph topology.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	15.6	16.0	0.1411
GGM	11.8	12.9	0.4167
RN	5.8	7.1	0.1780
DGE			
BN	11.3	15.8	0.0001
GGM	3.8	5.5	0.0045
RN	3.0	3.7	0.0729

Table B.29: **TP counts score. Observational versus Interventional data.** Gaussian data sets. V-structure graph topology.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	9.5	11.1	0.1757
GGM	9.6	9.6	1.0000
RN	9.3	7.1	0.0347
DGE			
BN	3.3	6.9	0.0134
GGM	5.1	4.1	0.1502
RN	5.1	1.7	0.0001

Table B.30: **TP counts score. Observational versus Interventional data.** Real cytoflow data sets.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	11.0	7.9	0.0102
GGM	12.0	5.2	0.0000
RN	6.9	2.0	0.0000
DGE			
BN	2.8	8.4	0.0001
GGM	5.1	3.7	0.0042
RN	0.8	0.0	0.0650

Table B.31: **TP counts score. Observational versus Interventional data.** Netbuilder data sets low noise level ($\sigma = 0.01$). Original graph topology.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	18.1	17.7	0.5180
GGM	16.5	13.6	0.0088
RN	16.8	8.0	0.0000
DGE			
BN	7.2	17.3	0.0000
GGM	5.5	5.4	0.3466
RN	5.5	1.2	0.0000

Table B.32: **TP counts score. Observational versus Interventional data.** Netbuilder data sets medium noise level ($\sigma = 0.1$). Original graph topology.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	15.5	16.0	0.6463
GGM	14.8	14.5	0.8408
RN	16.6	13.6	0.0397
DGE			
BN	4.1	14.1	0.0005
GGM	4.7	5.5	0.1411
RN	5.1	5.0	0.8798

Table B.33: **TP counts score. Observational versus Interventional data.** Netbuilder data sets high noise level ($\sigma = 0.3$). Original graph topology.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	9.8	7.2	0.0003
GGM	9.8	7.5	0.0003
RN	5.2	5.1	0.8171
DGE			
BN	3.9	5.6	0.0001
GGM	4.5	3.4	0.0338
RN	1.5	2.0	NA

Table B.34: **TP counts score. Observational versus Interventional data.** Netbuilder data sets low noise level ($\sigma = 0.01$). V-structure graph topology.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	15.6	15.7	0.7245
GGM	14.7	12.5	0.0020
RN	9.3	8.9	0.4468
DGE			
BN	12.0	15.4	0.0016
GGM	5.5	4.9	0.1411
RN	4.8	3.8	0.1078

Table B.35: **TP counts score. Observational versus Interventional data.** Netbuilder data sets medium noise level ($\sigma = 0.1$). V-structure graph topology.

Method	μ[TP|OBS]	μ[TP|INT]	p(.)
UGE			
BN	14.2	14.9	0.1991
GGM	13.2	12.5	0.3347
RN	13.6	12.8	0.2907
DGE			
BN	7.7	12.3	0.0047
GGM	5.5	5.5	NA
RN	5.5	5.5	0.2328

Table B.36: **TP counts score. Observational versus Interventional data.** Netbuilder data sets high noise level ($\sigma = 0.3$). V-structure graph topology.

B.5 Comparison: original vs. v-structure - AUC

This section of the supplementary material provides four tables, numbered from B.37 to B.40, with p-values we have used to compare the performance of Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN) on the original graph topology G_O and the v-structured graph topology G_V. We used this analysis for checking to which differences the inclusion of v-structures for the different methods leads. More precisely, we have been interested in answering the question whether the inclusion of v-structures leads to a larger improvement of the AUC scores for Bayesian networks than for the other two methods. To this end we have looked for each pair of methods M_i and M_j at the mean AUC score differences $AUC(M_i, G_O)$ - $AUC(M_j, G_O)$ and $AUC(M_i, G_V)$ - $AUC(M_j, G_V)$ to see whether the difference in performance alters between the two graph topologies. Then we computed the p-value of a two-sided two-sample t-test for the null hypothesis:

H_0: $\mu[AUC(M_i, G_O)\text{-}(M_j, G_O)] = \mu[AUC(M_i, G_V)\text{-}AUC(M_j, G_V)]$

against its two-sided alternative. Consequently, low p-values p(.) indicate that the mean AUC score differences alter on the two different graph topologies.

All four tables in this section have the same structure. After a row indicating the figure of merit (UGE or DGE) as well as the data set type (pure observational or interventional), there is one row for each of the three methods under comparison: Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN). In each of these rows the mean AUC scores for both directed acyclic graph topologies G_O and G_V as well as the p-values of the tests mentioned above can be found. Thereby the signs minus ('-') and plus ('+') have been added to the p-value entries to indicate whether for the method mentioned in the row the mean difference is higher for the graph topology with v-structures G_V ('+') or for the original graph G_O ('-'). So for example each plus sign ('+') indicates that the alteration of the differences introduced by v-structures is for

Method	μ[AUC|G$_O$]	μ[AUC|G$_V$]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.8848	0.9775	-	+0.0187	+0.2018
GGM	0.8814	0.8933	-0.0187	-	+0.8900
RN	0.6809	0.6987	-0.2018	-0.8900	-
DGE - Observational					
BN	0.7817	0.9487	-	+0.0049	+0.0168
GGM	0.7967	0.8257	-0.0049	-	+0.8908
RN	0.6407	0.6649	-0.0168	-0.8908	-
UGE - Interventional					
BN	0.9661	1.0000	-	-0.2143	-0.9997
GGM	0.8203	0.8878	-0.2143	-	+0.4727
RN	0.7097	0.7436	-0.9997	+0.4727	-
DGE - Interventional					
BN	0.9796	0.9976	-	-0.0259	-0.5804
GGM	0.7488	0.8220	+0.0259	-	+0.3743
RN	0.6631	0.7021	+0.5804	-0.3743	-

Table B.37: **AUC score. Cross method differences between the original graph topology G_O and v-structure topology G_V.** Gaussian data sets.

the benefit of the method mentioned in the row.

Method	μ[AUC|G_O]	μ[AUC|G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7901	0.7845	-	-0.0255	+0.2939
GGM	0.8143	0.8529	+0.0255	-	+0.0001
RN	0.7434	0.7170	-0.2939	-0.0001	-
DGE - Observational					
BN	0.6808	0.7354	-	+0.8580	+0.1256
GGM	0.7446	0.7927	-0.8580	-	+0.0000
RN	0.6893	0.6801	-0.1256	-0.0000	-
UGE - Interventional					
BN	0.7047	0.7102	-	-0.0034	+0.1316
GGM	0.7297	0.7900	+0.0034	-	+0.0007
RN	0.7537	0.7280	-0.1316	-0.0007	-
DGE - Interventional					
BN	0.8280	0.8413	-	-0.0111	+0.0183
GGM	0.6793	0.7258	+0.0111	-	+0.0008
RN	0.6973	0.6773	-0.0183	-0.0008	-

Table B.38: **AUC score. Cross method differences between the original graph topology G_O and v-structure topology G_V.** Netbuilder data sets low noise level ($\sigma = 0.01$).

Method	μ[AUC|G_O]	μ[AUC|G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9464	0.9887	-	-0.1423	+0.0000
GGM	0.8803	0.9567	+0.1423	-	+0.0005
RN	0.9323	0.8513	-0.0000	-0.0005	-
DGE - Observational					
BN	0.8572	0.9674	-	+0.2027	+0.0000
GGM	0.7957	0.8788	-0.2027	-	+0.0004
RN	0.8362	0.7915	-0.0000	-0.0004	-
UGE - Interventional					
BN	0.9346	0.9927	-	+0.1280	+0.0004
GGM	0.8300	0.8277	-0.1280	-	+0.1488
RN	0.8003	0.7483	-0.0004	-0.1488	-
DGE - Interventional					
BN	0.9678	0.9944	-	+0.2979	+0.0005
GGM	0.7574	0.7547	-0.2979	-	+0.1553
RN	0.7336	0.6931	-0.0005	-0.1533	-

Table B.39: **AUC score. Cross method differences between the original graph topology G_O and v-structure topology G_V.** Netbuilder data sets medium noise level ($\sigma = 0.1$).

Method	μ[AUC|G_O]	μ[AUC|G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9049	0.9332	-	+0.7264	+0.0021
GGM	0.8829	0.9038	-0.7264	-	+0.2127
RN	0.9163	0.9154	-0.0021	-0.2127	-
DGE - Observational					
BN	0.8208	0.8745	-	+0.5437	+0.1121
GGM	0.7979	0.8350	-0.5437	-	+0.2403
RN	0.8238	0.8447	-0.1121	-0.2403	-
UGE - Interventional					
BN	0.9053	0.9788	-	+0.0723	+0.0018
GGM	0.8571	0.8677	-0.0723	-	+0.2437
RN	0.8631	0.8214	-0.0018	-0.2437	-
DGE - Interventional					
BN	0.9219	0.9393	-	+0.7899	+0.1130
GGM	0.7776	0.7861	-0.7899	-	+0.2394
RN	0.7824	0.7500	-0.1130	-0.2394	-

Table B.40: **AUC score. Cross method differences between the original graph topology G_O and v-structure topology G_V.** Netbuilder data sets high noise level ($\sigma = 0.3$).

B.6 Comparison: original vs. v-strucuture network - TP

This section of the supplementary material provides four tables, numbered from B.41 to B.44, with p-values we have used to compare the performance of Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN) on the original graph topology G_O and the v-structured graph topology G_V. We used this analysis for checking to which differences the inclusion of v-structures for the different methods leads. More precisely, we have been interested in answering the question whether the inclusion of v-structures leads to a larger improvement of the sensitivity S outputed when accepting five false positive counts for Bayesian networks than for the other two methods. To this end we have looked for each pair of methods M_i and M_j at the mean sensitivity differences $S(M_i, G_O)$ - $S(M_j, G_O)$ and $S(M_i, G_V)$ - $S(M_j, G_V)$ to see whether the difference in performance alters between the two graph topologies. Then we computed the p-value of a two-sided two-sample t-test for the null hypothesis:
H_0: $\mu[S(M_i, G_O)\text{-}S(M_j, G_O)] = \mu[S(M_i, G_V)\text{-}AS(M_j, G_V)]$
against its two-sided alternative. Consequently, low p-values p(.) indicate that the mean sensitivities differences alter on the two different graph topologies.

All four tables in this section have the same structure. After a row indicating the figure of merit (UGE or DGE) as well as the data set type (pure observational or interventional), there is one row for each of the three methods under comparison: Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN). In each of these rows the mean sensitivity, when accepting five false positive counts, for both directed acyclic graph topologies G_O and G_V as well as the p-values p(.) of the tests mentioned above can be found. Thereby the signs minus ('-') and plus ('+') have been added to the p-value entries to indicate whether for the method mentioned in the row the mean difference is higher for

Method	μ[S|G$_O$]	μ[S|G$_V$]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7900	0.9750	-	+0.0381	+0.0028
GGM	0.7400	0.7375	-0.0381	-	+0.5753
RN	0.4050	0.3625	-0.0028	-0.5753	-
DGE - Observational					
BN	0.2450	0.7063	-	+0.0000	+0.0000
GGM	0.2350	0.2375	-0.0000	-	+0.8960
RN	0.1900	0.1875	-0.0000	-0.8960	-
UGE - Interventional					
BN	0.9250	1.0000	-	-0.2696	-0.6690
GGM	0.6600	0.8063	+0.2696	-	+0.7308
RN	0.3250	0.4437	+0.6690	-0.7308	-
DGE - Interventional					
BN	0.9200	0.9875	-	-0.8122	-0.3898
GGM	0.2600	0.3438	+0.8122	-	-0.0963
RN	0.0900	0.2313	+0.3893	+0.0963	-

Table B.41: **TP counts score. Cross method differences between the original graph topology G_O and v-structure topology G_V.** Gaussian data sets.

the graph topology with v-structures G$_V$ ('+') or for the original graph G$_O$ ('-'). So for example each plus sign ('+') indicates that the alteration of the differences introduced by v-structures is for the benefit of the method mentioned in the row.

Method	μ[S|G$_O$]	μ[S|G$_V$]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.5500	0.6125	-	+0.3647	+0.2907
GGM	0.6000	0.6125	-0.3647	-	+0.5051
RN	0.3450	0.3250	-0.2907	-0.5051	-
DGE - Observational					
BN	0.1400	0.2437	-	+0.1009	+0.3401
GGM	0.2550	0.2813	-0.1009	-	-0.4938
RN	0.0400	0.0938	-0.3401	+0.4938	-
UGE - Interventional					
BN	0.3950	0.4500	-	-0.0004	-0.0009
GGM	0.2600	0.4688	+0.0004	-	-0.7546
RN	0.1000	0.3187	+0.0009	+0.7546	-
DGE - Interventional					
BN	0.4200	0.4125	-	-0.3407	-0.0019
GGM	0.1850	0.2125	+0.3407	-	-0.0000
RN	0.0000	0.1250	+0.0019	+0.0000	-

Table B.42: **TP counts score. Cross method differences between the original graph topology G_O and v-structure topology G_V.** Netbuilder data sets low noise level ($\sigma = 0.01$).

Method	μ[S|G$_O$]	μ[S|G$_V$]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9050	0.9750	-	-0.4794	+0.0000
GGM	0.8250	0.9187	+0.4794	-	+0.0000
RN	0.8400	0.5813	-0.0000	-0.0000	-
DGE - Observational					
BN	0.3600	0.7500	-	+0.0003	+0.0002
GGM	0.2750	0.3438	-0.0003	-	+0.0729
RN	0.2750	0.3000	-0.0002	-0.0729	-
UGE - Interventional					
BN	0.8850	0.9812	-	-0.9014	-0.3257
GGM	0.6800	0.7813	+0.9014	-	-0.1904
RN	0.4000	0.5563	+0.3257	+0.1904	-
DGE - Interventional					
BN	0.8650	0.9625	-	+0.0750	-0.0395
GGM	0.2700	0.3063	-0.0750	-	-0.0055
RN	0.0600	0.2375	+0.0395	+0.0055	-

Table B.43: **TP counts score. Cross method differences between the original graph topology G_O and v-structure topology G_V.** Netbuilder data sets medium noise level ($\sigma = 0.1$).

Method	μ[S|G$_O$]	μ[S|G$_V$]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7750	0.8875	-	+0.5750	+0.0859
GGM	0.7400	0.8250	-0.5750	-	+0.0889
RN	0.8300	0.8500	-0.0859	-0.0889	-
DGE - Observational					
BN	0.2050	0.4813	-	+0.0446	+0.0224
GGM	0.2350	0.3438	-0.0446	-	+0.1413
RN	0.2550	0.3125	-0.0224	-0.1413	-
UGE - Interventional					
BN	0.8000	0.9313	-	+0.1437	+0.8220
GGM	0.7250	0.7813	-0.1437	-	-0.2778
RN	0.6800	0.8000	-0.8220	+0.2778	-
DGE - Interventional					
BN	0.7050	0.7688	-	-0.9577	-0.7767
GGM	0.2750	0.3438	+0.9577	-	-0.3466
RN	0.2500	0.3438	+0.7767	+0.3466	-

Table B.44: **TP counts score. Cross method differences between the original graph topology G_O and v-structure topology G_V.** Netbuilder data sets high noise level ($\sigma = 0.3$).

Bibliography

Atkins, P. W. (1986). *Physical Chemistry*. Oxford University Press, Oxford, 3rd edition.

Bernard, A. and Hartemink, A. J. (2005). Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. In *Pacific Symposium on Biocomputing*, pages 459–470, New Jersey. World Scientific.

Bing, N. and Hoeschele, I. (2005). Genetical genomics analysis of a yeast segregant population for transcription network inference. *Genetics*, 170:533–542.

Bonetta, L. (2005). Flow cytometry smaller and better. *Nature Methods*, 2:785–795.

Briggs, G. and Haldane, J. (1925). A note on the kinetics of enzyme action. *Biochemical Journal*, 19:339–339.

Buntine, W. L. (1991). Theory refinement of Bayesian networks. In *Uncertainty in Artificial Intelligence*.

Butte, A. S. and Kohane, I. S. (2000). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing*, 2000:418–429.

Butte, A. S. and Kohane, I. S. (2003). Relevance networks: A first step toward finding genetic regulatory networks within microarray data. In Parmigiani, G.,

Garett, E. S., Irizarry, R. A., and Zeger, S. L., editors, *The Analysis of Gene Expression Data*, pages 428–446, New York. Springer.

Chen, T., He, H. L., and Church, G. M. (1999). Modeling gene expression with differential equations. *Pacific Symposium on Biocomputing*, 4:29–40.

Chickering, D. M. (2002). Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498.

Claverie, J.-M. (2005). Fewer genes, more noncoding RNA. *Science*, 309(5740):1529–1530.

Cooper, G. and Glymour, C. (1999). *Computation, Causation, and Discovery.* MIT Press, Cambridge, MA.

Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91:883–904.

Crick, F. H. (1970). Central dogma of molecular biology. *Nature*, 227:561–563.

De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103.

D'haeseleer, P., Liang, S., and Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726.

Dougherty, M. K., Müller, J., Ritt, D. A., Zhou, M., Zhou, X. Z., Copeland, T. D., Conrads, T. P., Veenstra, T. D., Lu, K. P., and Morrison, D. K. (2005). Regulation of Raf-1 by direct feedback phosphorylation. *Molecular Cell*, 17:215–224.

Eaton, D. and Murphy, K. (2007). Bayesian structure learning using dynamic programming and MCMC. In *Proceedings of the Twenty-Third Confererence on Uncertainty in Artificial Intelligence (UAI 2007)*.

Edwards, D. M. (2000). *Introduction to Graphical Modelling*. Springer Verlag, New York.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95:14863–14868.

Friedman, N. and Koller, D. (2003). Being Bayesian about network structure. *Machine Learning*, 50:95–126.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620.

Friedman, N., Murphy, K., and Russell, S. (1998). Learning the structure of dynamic probabilistic networks. In Cooper, G. F. and Moral, S., editors, *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 139–147, San Francisco, CA. Morgan Kaufmann.

Geiger, D. and Heckerman, D. (1994). Learning Gaussian networks. In de Mantaras, R. L. and Poole, D., editors, *Uncertainty in Artificial Intelligence*, pages 235–243, San Francisco, CA. Morgan Kaufmann.

Geyer, C. J. (1991). Markov chain monte carlo maximum likelihood. In M, K. E., editor, *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 156–163, Interface Foundation. Fairfax Station.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). Strategies for improving MCMC. In Gilks, W. R. and Roberts, G. O., editors, *Markov Chain*

Monte Carlo in Practice, pages 89–114, Suffolk. Chapman & Hall. ISBN 0-412-05551-1.

Giudici, P. and Castelo, R. (2003). Improving Markov chain Monte Carlo model search for data mining. *Machine Learning*, 50:127–158.

Grzegorczyk, M. (2006). *Comparative Evaluation of different Graphical Models for the Analysis of Gene Expression Data*. PhD thesis, Department of Statistics of the University Dortmund, Dortmund.

Harbison, C. T., Gordon, B. D., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J.-B., Reynolds, D. B., Yoo, J., Jennings, E. G., Zeitlinger, J., Pokholok, D. K., Kellis, M., Rolfe, A. P., Takusagawa, K. T., Lander, E. S., Gifford, D. K., Fraenkel, E., and Young, R. A. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104.

Hartemink, A. J. (2001). *Principled computational methods for the validation and discovery of genetic regulatory networks*. PhD thesis, MIT.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

Heckerman, D. (1994). Learning Gaussian networks. Technical Report MSR-TR-94-10, Microsoft Research, Redmond, Washington.

Heckerman, D. (1995). A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington,.

Heckerman, D. (1999). A tutorial on learning with Bayesian networks. In Jordan, M. I., editor, *Learning in Graphical Models*, Adaptive Computation and Machine Learning, pages 301–354, Cambridge, Massachusetts. MIT Press.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:245–274.

Herzenberg, L. A., Parks, D., Sahf, B., Perez, O., Roederer, M., and Herzenberg, L. A. (2002). The history and future fo the fluorescence activated cell sorter and flow cytometry: A view from stanford. *Clinical Chemistry*, 48(10):1819–1827.

Hill, A. (1910). The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *Journal of Physiology*, 40:4–7.

Huelsenbeck, J. P. and Ronquist, F. (2001). Mr Bayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17:754–755.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19:2271–2282.

Husmeier, D. and Werhli, A. V. (2007). Bayesian integration of biological prior knowledge into the reconstruction of gene networks with Bayesian network. *To appear in CSB2007*.

Ideker, T., Ozier, O., Schwikowski, B., and Siegel., A. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18 Suppl 1:S233–40.

Imoto, S., Higuchi, T., Goto, T., and Miyano, S. (2006). Error tolerant model for incorporating biological knowledge with expression data in estimating gene networks. *Statistical Methodology*, 3(1):1–16.

Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., and Miyano, S. (2003a). Combining microarrays and biological knowledge for estimating gene networks

via Bayesian networks. *Proceedings IEEE Computer Society Bioinformatics Conference*, (CSB'03):104–113.

Imoto, S., Kim, S., Goto, T., Aburatani, S., Tashiro, K., Kuhara, S., and Miyano, S. (2003b). Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *Journal of Bioinformatics and Computational Biology*, 1(2):231–252.

Kanehisa, M. (1997). A database for post-genome analysis. *Trends Genet*, 13:375–376.

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28:27–30.

Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K., Itoh, M., Kawashima, S., Katayama, T., Araki, M., and Hirakawa, M. (2006). From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34:D354–357.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Laskey, K. B. and Myers, J. W. (2003). Population Markov chain Monte Carlo. *Machine Learning*, 50(1-2):175–196.

Ledoit, O. and Wolf, M. (2004). A well conditioned estimator for large-dimensional covariance matrices. *Journal of Mutivariate Analysis*, 88:365–411.

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J. B., Volkert, T. L., Fraenkel, E., Gifford, D. K., and Young, R. A. (2002). Transcriptional regulatory networks in *Saccharomyces cerevisiae* . *Science*, 298(5594):799–804.

Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review*, 63:215–232.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

Michaelis, L. and Menten, M. (1913). Die kinetik der invertinwirkung. *Biochemische Zeitschrift*, 49:333–369.

Murphy, K. and Milan, S. (1999). Modelling gene expression data using dynamic Bayesian networks. Technical report, MIT artificial intelligence laboratory.

Murphy, K. P. (2001). An introduction to graphical models. Technical report, MIT, Artificial Inteligence Laboratory.

Nariai, N., Kim, S., Imoto, S., and Miyano, S. (2004). Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks. *Pacific Symposium on Biocomputing*, 9:336–347.

Nariai, N., Tamada, Y., Imoto, S., and Miyano, S. (2005). Estimating gene regulatory networks and protein-protein interactions of *Saccharomyces cerevisiae* from multiple genome-wide data. *Bioinformatics*, 21(Suppl.2):ii206–ii212.

Pearl, J. (2000). *Causality: Models, Reasoning and Intelligent Systems*. Cambridge University Press, London, UK.

Pearson, H. (2006). What is a gene? *Nature*, 441:399–401.

Perez, O. D. and Nolan, G. P. (2002). Simultaneous measurement of multiple active kinase states using polychromatic flow cytometry. *Nature Biotechnology*, 20:155–162.

Pournara, I. and Wernisch, L. (2007). Factor analysis for gene regulatory networks and transcription factor activity profiles. *BMC Bioinformatics*, 8(61).

Pournara, I. V. (2005). *Reconstructing gene networks by passive and active Bayesian learning*. PhD thesis, Birbeck College, University of London.

Sabatti, C. and James, G. M. (2005). Bayesian sparse hidden components analysis for transcription regulation networks. *Bioinformatics*, 22(6):739–746.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529.

Schäfer, J. and Strimmer, K. (2005a). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764.

Schäfer, J. and Strimmer, K. (2005b). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4. Article 32.

Schena, M., Shalon, D., Davis, R., and Brown, P. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470.

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13:2498–2504.

Sing, G., Werhli, A. V., Dickinson, P., Ross, A. J., Sorokin, A., Moodie, S., Cho, L., Frankenberg, T., kun Phng, L., Xu, Z., Yao, L.-C., Robertson, K., Forster, T., Livingston, A., Roy, D., Beattie, J., Selkov, A., Angulo, A., Husmeier, D., Goryanin, I., and Ghazal, P. (2006). Application of an interferon logic interaction network to the analysis of macrophage response to infection. Unpublished paper.

Smith, V. A., Jarvis, E. D., and Hartemink, A. J. (2002). Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, 18:S216–S224. (ISMB02 special issue).

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol Biol Cell*, 9(12):3273–3297.

Spirtes, P., Glymour, C., and Scheines, R. (2001). *Causation, Prediction, and Search*. Springer Verlag, New York.

Tamada, Y., Bannai, H., Imoto, S., Katayama, T., Kanehisa, M., and Miyano, S. (2005). Utilizing evolutionary information and gene expression data for estimating gene networks with Bayesian network models. *Journal of Bioinformatics and Computational Biology*, 3(6):1295–1313.

Tamada, Y., Kim, S., Bannai, H., Imoto, S., Tashiro, K., Kuhara, S., and Miyano, S. (2003). Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics*, 19:ii227–ii236.

Tian, J. and Pearl, J. (2001). Causal discovery from changes: a bayesian approach.

Tu, B. P., Kudlicki, A., Rowicka, M., and Mcknight, S. L. (2005). Logic of the yeast metabolic cycle: Temporal compartmentalization of cellular processes. *Science*, 310(5751):1152–1158.

Verma, T. and Pearl, J. (1991). Equivalence and synthesis of causal models. In *Proceedings of the 6th Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)*, pages 255–270, New York, NY. Elsevier Science.

Watson, J. D. and Crick, F. H. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737–738.

Watson, J. D. and Crick, F. H. (1958). On protein synthesis. *The Symposia of the Society for Experimental Biology*, 12:138–163.

Werhli, A. V., Grzegorczyk, M., and Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. *Bioinformatics*, 22(20):2523–2531.

Werhli, A. V. and Husmeier, D. (2007). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, 6(1):Article 15.

Wernisch, L. and Pournara, I. (2004). Reconstruction of gene networks using bayesian learning and manipulation experiments. *Bioinformatics*, 20:2934–2942.

Ying, S.-Y. and Lin, S.-L. (2004). Intron-derived microRNAs - fine tuning of gene functions. *Gene*, 342(1):25–28.

Yuh, C. H., Bolouri, H., and Davidson, E. H. (1998). Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902.

Yuh, C. H., Bolouri, H., and Davidson, E. H. (2001). Cis-regulatory logic in the endo16 gene: switching from a specification to a differentiation mode of control. *Development*, 128:617–629.

Zak, D. E., Doyle, F. J., Gonye, G. E., and Schwaber, J. S. (2001). Simula-

tion studies for the identification of genetic networks from cDNA array and regulatory activity data. pages 231–238, Pasadena, CA.